

⑫

EUROPEAN PATENT APPLICATION

⑲ Application number: **87311086.0**

⑤① Int. Cl.4: **G06F 3/033**

⑳ Date of filing: **16.12.87**

③① Priority: **22.12.86 US 945431**

④③ Date of publication of application:
20.07.88 Bulletin 88/29

⑤④ Designated Contracting States:
DE FR GB

⑦① Applicant: **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504(US)

⑦② Inventor: **Ballou, Richard V.**
8408 Linden Way
Lake Worth Florida 33467(US)
Inventor: **Belyeu, Stanley M.**
1233 NW 14th Street
Boca Raton Florida 33432(US)
Inventor: **Boscove, Joseph A.**
4425 Tranquility Drive
Highland Beach Florida 33431(US)
Inventor: **Kurtz, Hobart L.**
425 N.W. 11th Street
Boca Raton Florida 33432(US)
Inventor: **Langer, Peter**
1826 N.W. 88 Way
Coral Springs Florida 33065(US)
Inventor: **McNeill, Andrew B.**
181 N.W. 41st Way
Deerfield Beach Florida 33442(US)
Inventor: **Reid, Bernard M.**
8433 Forest Hills Drive
Coral Springs Florida 33065(US)
Inventor: **Rodriguez, Herman**
Route 5 Box 75
Live Oak Florida 32060(US)

⑦④ Representative: **Atchley, Martin John**
Waldegrave
IBM United Kingdom Limited Intellectual
Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

EP 0 274 879 A2

⑤④ **System for generating frames of information for display on a display system.**

⑤⑦ The invention relates to a system for generating frames of information for displaying on a display systems, of the type including a display unit (36), a display controller and a control program for operating the display controller, comprising a data processor including associated storage and display means, and a generating program for the data processor for selectively generating frame data, and editing and storing the data, the data being representative of graphics elements, symmols, icons and animation required for each frame to be generated.

According to the invention the system is characterised in that the generating program operates independently of the control program, whereby the frame data can be generated without altering the control program.

SYSTEM FOR GENERATING FRAMES OF INFORMATION FOR DISPLAY ON A DISPLAY SYSTEM

This invention relates to display systems which can be used for example in vehicles, particularly automotive vehicles. Automobile manufacturers are adding display systems including Graphic Display Centres (GDC) to their high technology vehicles. This will be a continuing trend over time. The GDC contains a display unit and a mixture of hard (requiring physical movement for operation) and soft (requiring only touch for operation) switches. Use of a display unit, like a CRT, in the display system offers the capability to provide a large amount of information to the vehicle operator. The information is displayed on the screen of the display unit in successive sets of information. In the present description each set of information displayed is termed a "frame" or in some places a "screen". The hard switches are used to control the display of the basic frames of information. A touch panel which overlays the display unit provides frame related "soft" switches whose functions are unique to the frame being displayed. Many of the soft switches, when actuated, cause sub-frames of the basic frame to be displayed. Sub-frames can in turn cause other sub-frames to be invoked. Each vehicle using a GDC will usually contain over 100 different frame layouts. Frames contain a multiplicity of different types of display elements such as icons, Symbols, lines, boxes, circles, text, etc.

The following references are of interest and are representative of the known arrangement.

1. Article: "Evolving Microprocessors Which Better Meet The Needs Of Automotive Electronics" John Marley, Proceedings of the IEEE, Vol. 66, No. 2, February 1978, Pages 142-150.
2. Article: "Evaluating Pictographic Symbols: An Automotive Application" Paul Green et al, Human Factors, Vol 20, No. 1, February 1978, Pages 103-114.
3. Article: "Simplification Of System Inputs And Outputs For MPU Control Units" John Marley, Society of Automotive Engineers, Technical Paper Series No. 780123, February 27-March 3, 1978, Pages 1-6.
4. Article: "An Innovative Approach In Designing An Electronic Instrument Cluster" Frederick O. R. Miesterfeld, Society of Automotive Engineers, Congress and Exposition Detroit, MI Technical Paper Series No. 810308, February 23-27, 1981, Pages 101-106.
5. Article: "Multiprocessor Implementation Of The Logic Function Of A Multiplexed Wiring System For Automotives" N.C.G.N. Preston et al, IEE Proc., Vol. 129, Pt. E, No. 6, November 1982, Pages 223-228.
6. Article: "Touch Sensitive Displays For Operation And Control" Michael Menkin et al, Electronic Engineering, September 1983, Pp. 177-180.
7. Article: "More Electronics In Detroit's 1985 Models" Ronald K. Jurgen, Senior Editor, IEEE Spectrum, October 1984, Pages 54-60.
8. Article: "Micros Hit The Road", David Whiteside, Datamation, October 1, 1984, Pages 22-26.
9. Article: "GM Tests Video Display In 1984 Production Car", David M. Weber, Electronics Week, November 12, 1984, Pages 28-29.
10. Article: "Automakers Shift To Processors" Richard Valentine et al, Electronics Week, December 10, 1984, Pages 61-65.
11. Article: "An Interactive Reconfigurable Display System For Automotive Instrumentation" Jesse Ortega et al, SAE Society of Automotive Engineers Congress and Exposition SAE, Warrendale, PA February 24-28, 1986, Pages 1-14. Technical Paper Series No. 860173.
12. Article: "Motoring At 6MHz" John Eckhouse, PC World, April 1986, Pages 108-109.
13. Brochure: "High Technology From Buick" Ref. Buick Riviera Graphic Control Center Published by Product Information Buick Motor Division, (General Motors Corporation, USA).

In known frame generating systems the frames to be used in a GDC are first designed and displayed on a computer screen. Either the full computer screen or a portion of the screen is used to simulate the vehicle display unit. The aspect ratio and colour of the computer screen is different from that of the vehicle display unit. A time consuming process is required for a frame to be viewed on the actual vehicle screen. This requires processing of the developed frames using a series of programs that allow generating a frame image that can be burned into an EPROM. Once generated, the EPROM must be installed and run on a bench test setup that contains the actual vehicle display unit. When the frame is viewed on the actual vehicle display unit, changes may be necessary due to the vehicle display unit being different from the display unit that was used to develop the frame. Changes necessitate going back to the computer, modifying the frame and going through the long and tedious procedure of burning an EPROM and displaying the frame on a bench test setup.

The invention of the present application relates to display systems and more particularly to a more efficient system for creating and utilising the display frames used therein.

As noted, automobile companies are making greater use of display systems in their vehicles. Such a

system display typically includes a controller and provides for inputs by the vehicle operator as well as displaying a variety of frames of information. Thus, a greater need exists for being able to create and update the frames for use in such vehicles.

5 The object of the present invention is to provide an improved system for generating frames of information for displaying on a display system.

The invention relates to a system for generating frames of information for displaying on a display system, of the type including a display unit, a display controller and a control program for operating the display controller, comprising a data processor including associated storage and display means, and a
10 generating program for the data processor for selectively generating frame data, and editing and storing the data, the data being representative of graphics elements, symbols, icons and animation required for each frame to be generated.

According to the invention the system is characterised in that the generating program operates independently of the control program, whereby the frame data can be generated without altering the control
program.

15 The system provides a method and capability to design, create and update automotive display screens, and is able to include any Controller function, without rewriting or modifying any software.

In the system described herein, Interactive Graphic display frames are displayed in a manner which does not require the control program to be changed when a frame design is changed. This arrangement is made possible by a table based frame data structure and control program architecture that allows the
20 control program to efficiently identify and use each frame's design data to update the display and perform actions in response to recognised stimuli such as touch key activation, communication message data, internal conditions, etc.

This invention also eliminates the problem of not being able to view automobile frames on the actual auto display as they are created or modified.

25 In order that the invention may be more readily understood an embodiment will now be described with reference to the accompanying drawings, in which,

Fig. 1 illustrates a typical frame of information for displaying on a display unit,

Fig. 2 illustrates the interior of a typical vehicle,

Fig. 3 is a block diagram of a vehicle display system,

30 Fig. 4 is a block diagram of a vehicle display system controller that is used in the system of Fig. 3,

Fig. 5 is flow chart of the frame development process,

Fig. 6 is a pictorial view of a Graphics Development Instrument System (GDIS) work station,

Fig. 7 shows a list of the GDIS work station hardware,

35 Fig. 8 is a more detailed block diagram of the GDIS work station of Figs. 6 and 7,

Fig. 9 illustrates a power card used in the work station of Figs. 6-8,

Fig. 10 shows details of the video card used in the work station of Figs. 6-8,

Fig. 11 is an overview of the GDIS work station software,

Fig. 12 is an overview of the functions of the vehicle display frame generator program,

Fig. 13 illustrates typical graphics elements used by the GDIS system,

40 Fig. 14 illustrates the merging of several vehicle display system frames to create a vehicle frame set via Post-Processor and Link programs,

Fig. 15 represents the use of the vehicle frame set by the display system controller to generate frame images, automobile functions and animation,

Fig. 16 is a high level diagram of communications of a host computer with the GDIS system,

45 Fig. 17 shows a typical GDIS network,

Fig. 18 is a flow chart of the GDIS control program,

Fig. 19 is a flow chart of the GDIS Post-Processor program,

Fig. 20 is a block diagram of the frame data structure generated by the Post-Processor program,

Fig. 21 is a flow chart of the Touch function handler in the controller software,

50 Fig. 22 is a flow chart of the Poll function handler in the controller software,

Fig. 23 is a flow chart of the Animation handler in the display system controller software, and

Fig. 24 is a flow chart of the Screen Loader routine which is also part of the vehicle display system controller software.

55 A Graphics Development Instrument System (GDIS) for use in developing frames of information for display on a vehicle display system will now be described.

The use of the GDIS to be described is best understood in conjunction with the Vehicle Display System

described in the copending European patent Application, [] which corresponds to United States Patent Applicatoin No 945053, filed on 22 December 1986.

The following abbreviations are occasionally used in this description.

5

10

15

20

25	Abbreviations	Definition
	CRT	Cathode Ray Tube
	DIM IN	Dimming Control line for CRT
30	CRT CXEDRAW	Graphics Language Routines
	DCS	Display Controller System
	EPROM	Erasable Programmable Read Only
35		Memory
	EEROM or	Electrically Erasable Read
	EEPROM	Only Memory

40

45

50

55

	GDIS	Graphics Development Instrument
5		System
	GDL	Graphics Design Language
	GND	Ground
	HSYNC	Horizontal Synchronization
10	IGN1	Ignition Switch 1
	IGN2	Ignition Switch 2
	INITKEY	Initialize Key Number
15	KB	Kilobyte PWR
	Power	
	RAM	Random Access Memory
	ROM	Read Only Memory
20	RTN	Return
	SDL	Serial Data Link
	TCON	Not Touch Controller On
25	TEMP	Temperature
	UART	Universal Asynchronous Receiver
		Transmitter
30	V SYNC	Vertical Synchronization
	VBAT	Battery Voltage (+12)
	VBATF	Filtered Battery Voltage
	VGA	Video Gate Array
35	VOL	Volume

The following circuit blocks are included in the drawings:

40	<u>Block</u>	<u>Function/Description</u>
45	80C88	Intel Microprocessor (CMOS) - 8088
	80C31	Intel Microcontroller (CMOS) - 8031
50	82C55	Intel Programmable Interface Adapter - 8255

A Vehicle Display System, such as that described in the above Patent Application, consists of a CRT
 55 display unit and monitor electronics, a touch switch matrix and decoding electronics, and the display
 system controller electronics. The Vehicle Display System controller is a computerised unit which controls
 the operation of a touch sensitive video display. The controller is intended to be connected to a touch
 activated colour or monochrome CRT display and to be used in an automobile to provide for the control and

monitoring of various vehicle functions. The touch sensitive display system serves as a centralised command centre for control of the vehicle's climate control system, radio, and trip computer. Several convenience functions are also provided.

The display system may also visually indicate the status of the vehicle's electrical and mechanical systems. The controller continuously monitors these systems via serial data links and discrete interconnections with the vehicle's electrical system. It will provide appropriate messages to the driver of the vehicle in the event of a condition which requires the driver's attention. An automatic maintenance log is also provided to remind the driver when scheduled service is required and to record the service mileage and date of service permanently. An on-line owner's manual may also be displayed to provide information such as filter types/sizes, fluid types/capacities, etc.

Some other features are listed below:

- Automatic Cellular Telephone Dialler and Directory
- Electronic Compass Display
- Appointment Calendar with Automatic Reminder
- Calculator

As previously noted, automobile manufacturers are adding Graphic Display Centers (GDC) to their high technology vehicles. The GDC contains a display and a mixture of hard and soft switches. Use of a display unit, like a CRT, provides the capability to provide much more information to the vehicle operator. The hard switches are used to display the basic frames of information. A Touch Panel which overlays the display unit provides for frame related "soft" switches whose functions are unique to the frame being displayed. Many of the soft switches when actuated cause sub-frames of the basic frame to be displayed. Sub-frames can in turn cause other sub-frames to be invoked. Each vehicle using a Display System will contain over 100 frame definitions. Frames contain a multiplicity of different types of display elements such as Icons, Symbols, lines, boxes, circles, text, etc. An element is further classified into one of two categories, static or dynamic. A static element is one which never changes while the frame is displayed. A dynamic element is one which will change as a result of some form of stimulus such as the activation of one of the soft switches on the panel, a message from some other controller, output from a sensor, etc. A frame definition must therefore include, in addition to the graphic elements, specifications of the functions to be supported. Up to now, each of the dynamic functions defined within each of the frames is supported by a unique custom program module. Given the number of frames per vehicle type, the number of vehicle types to be supported and the number of programs associated with dynamic elements, the amount of resources required to develop frames for a vehicle Display System has become exorbitant. The GDIS describe herein solves this problem.

Known systems for developing frames for display systems require design, coding and testing controller software when display frames are created or modified. The GDIS described herein allows for rapid update of a vehicle's frame set, so that the testing efforts of the frame designer are concentrated on the frame contents and its interaction with the operator, and not on the software that drives the system.

This system also creates an indexed table structure for the Graphic Frame Data to allow the Control program to quickly identify and use the elements which compose the frame without having to modify the Control program when a frame design is changed.

Thus, the GDIS allows the frame designer to incorporate any controller function on any frame, and to create the EPROM's of those frames, without having to write or modify any code. This feature allows the designer to change quickly any button or graphic definition of any frame. However, the key attribute of the GDIS is that it allows the frame designer to specify freely which controller functions are made available on a frame, and, after some human factors testing, he can add or subtract functions or elements from it as needed. This is accomplished using first the GDIS program to draw the frame. Its output consists of graphics language statements representing the images, button definitions and Controller functions of the frame.

Controller functions are those features and options of a car that can be associated with graphics elements. For example, the radio "ON" status of a radio is defined as a Controller function that can be included in any frame definition. This feature of the GDIS allows the automobile display frame designer to experiment with several screen designs, including touch button definitions, in order to achieve the best results.

The language output of the GDIS utility program is run through a Post-Processor program to complete the definition of a vehicle's frame set. The output of the Post-Processor program is assembled using the Macro Assembler, in order to create the frame definitions the Controller will use in the vehicle. A series of

conversion steps produces the EPROM that is installed in the vehicle.

Also described herein is the architecture of the data structure and Control program that provides the ability to display interactive Graphic display frames in an efficient and structured manner. This architecture allows the frame designs (frame data) to be changed without requiring corresponding changes in the program that controls the display of the frames. The indexed structure of the frame data allows the Control
 5 program to interpret and manipulate the Graphics elements that compose each frame efficiently to achieve the desired visual effects. The effects can include normal updating of display information due to changes in variable data as well as simple forms of animation.

A typical display frame is shown in Fig. 1. The GDIS described supports actions which are to occur
 10 when a stimulus is detected by the Control program. In the case of the Controller, the stimulus can take the form of a Touch Switch activation, an internal condition, or a message being received from one of the Serial Data Links that provide communications and that are supported by the Controller.

All Graphic frames displayed by the Vehicle Display System are based on Graphic Frame Data contained in a ROM memory. The frame data is created with the Graphics Development Instrument System.
 15 The Table based structure of the frame data allows the frame designs to be changed without changing the Control program. The Table based architecture is an important feature of this invention. The Graphics Frame Data is compacted to provide maximum use of available ROM memory space. This data contains tables which point to and describe graphic elements and animation sequences, specify Touch switch function assignments, and indicate which Polling functions are used by each frame.

The compacted Graphics Frame Data is converted on demand to software function calls by the
 20 Graphics Language Interpreter (part of the Control program). The Graphics Language Interpreter controls the drawing of lines, rectangles, filled rectangles, circles, filled circles, filled triangles, bar graphs, three character fonts, and custom two colour symbols and icons. Also supported is the ability to display symbols or icons in a sequence to allow the creation of simple animation effects.

Functions which occur as a direct result of a Touch switch activation are performed by execution of a
 25 program subroutine written specifically for each function. The collection of subroutines to handle all the Touch activated functions allowed by the Controller and a calling routine to perform the Touch Switch to Function conversion is known as the Touch Function Handler. The Touch Function Handler is responsible for performing the desired action and for displaying the appropriate graphical response on the display.
 30 Touch functions are specified by Dynamic Element Names assigned to Touch elements by the Graphics Development Instrument when the frame is created. The graphical response to a Touch function is specified by the Dynamic Element Names assigned to the frame's graphical elements when the frame is designed.

Graphical response to changes which are not the direct result of Touch Switch activation are known as
 Poll functions. These functions include the updating of values such as the time of day, trip odometer, etc,
 35 and the updating of certain indicators such as the stereo radio status. The Poll Function Handler software updates the Display frame several times each second. Each time the control program calls the Poll Function Handler, a Poll function is executed from the list in the displayed frame's Poll function data table. Poll functions are specified by unique Dynamic Element Names assigned to Graphic Elements when the Display frame is designed.

The display and speed control of graphical animation is performed by Animation Handler software. Each
 40 kind of animation sequence is handled by an associated program subroutine. Each subroutine is called by a Calling program driven by the currently displayed frame's Animation function data table. Each time the Animation Handler is called by the Control program all animation sequences on the frame are updated. The Animation sequence parameters are also specified via the GDIS.

Three text character fonts are supported by the Controller to allow the display of text and numerical
 45 information. The standard ASCII character set is supported by the Graphics Language Interpreter. The font data may be changed to support different character sets. For example, three different character sets are implemented. The sets provide characters of 8×8 , 8×14 , and 16×128 pixels (picture elements) in size. The character background and foreground colours and font are specified at the time the frame is designed with
 50 the Graphics Development Instrument System.

Frame Data Creation

55 For each display frame that is created with the Graphics Development Instrument System (GDIS), a Graphic Design Language (GDL) text file which describes the frame design is generated. The GDL text file for the frame shown in Fig. 1 is listed below.

AUTHOR John Doe
 DATE 09-01-1986
 REF 111
 STYPE 2
 5 FLAGS 2
 BCOLOUR 0
 ECOUNT 55
 FBOX S, 125, 145, 183, 175, 14, NULL; 80, 69, 4, 59, 31
 BOX S, 100, 50, 215, 110, 2, NULL; 80, 69, 1, 116, 61
 10 BOX S, 8, 128, 57, 191, 14, NULL; 80, 68, 3, 50, 64
 BOX S, 99, 49, 216, 111, 2, NULL; 80, 69, 1, 118, 63
 ICON S, DARROW, 9, 167, 2, 0, NULL; 80, 68, 5
 ICON S, UPARROW, 9, 129, 2, 0, NULL; 80, 68, 4
 LINE S, 0, 120, 319, 120, 15, NULL;
 15 TEXT S, 20, 155, 15, 0, 2, "SET", NULL; 80, 69, 1
 TEXT S, 47, 152, 15, 0, 1, "O", NULL; 80, 69, 1
 LINE S, 0, 43, 319, 43, 15, NULL;
 TEXT S, 172, 147, 0, 14, 1, "O", NULL; 80, 69, 1
 TEXT S, 22, 74, 15, 0, 2, "VOL", NULL; 80, 65, 3
 20 BOX S, 8, 49, 57, 111, 2, NULL; 80, 68, 3, 50, 63
 ICON S, UPARROW, 9, 50, 14, 0, NULL; 80, 68, 2
 ICON S, DARROW, 9, 87, 14, 0, NULL; 80, 68, 3
 TEXT S, 111, 180, 15, 0, 2, "INSIDE TEMP", NULL; 80, 69, 1
 ICON D, VETBUTT, 264, 148, 14, 0, HVACOFF; 80, 69, 5
 25 TEXT D, 277, 254, 14, 0, 2, "PWR", HVACOFF; 80, 69, 5
 ICON D, VETBUTR, 264, 148, 14, 0, HVACONI; 80, 69, 5
 TEXT D, 277, 154, 0, 14, 2, "PWR", HVACONI; 80, 69, 5
 TEXT D, 135, 147, 0, 14, 3, "75", STEMPF; 80, 69, 1
 TEXT D, 172, 158, 0, 14, 2, "C", TMPUNITC; 80, 69, 1
 30 TEXT D, 172, 158, 0, 14, 2, "F", TMPUNITC; 80, 69, 1
 ICON D, VETBUTT, 264, 68, 2, 0, RAD-OFF; 80, 69, 3
 TEXT D, 277, 74, 2, 0, 2, "PWR", RAD-OFF; 80, 69, 3
 ICON D, VETBUTR, 264, 68, 2, 0, RAD-ONI; 80, 69, 3
 TEXT D, 277, 74, 0, 2, 2, "PWR", RAD-ONI; 80, 69, 3
 35 TEXT D, 118, 57, 0, 0, 3, "12345", FREQBLNK; 80, 69, 1
 TEXT D, 119, 88, 0, 0, 2, "FM", AMFMBLNK; 80, 69, 1
 TEXT D, 118, 57, 15, 0, 3, "1420", STAT-AM; 80, 69, 1
 TEXT D, 119, 88, 15, 0, 2, "AM", AM-IND; 80, 69, 1
 TEXT D, 118, 57, 15, 0, 3, "107.9", STAT-FM; 80, 69, 1
 40 TEXT D, 119, 88, 15, 0, 2, "FM", FM-IND; 80, 69, 1
 TEXT D, 149, 88, 0, 0, 2, "STEREO", STEROFF; 80, 69, 1
 TEXT D, 149, 88, 15, 0, 2, "STEREO", STERONI; 80, 69, 1
 TEXT D, 110, 10, 11, 0, 3, "12:28", CLOCK; 80, 65, 1
 TEXT D, 195, 21, 10, 0, 2, "AM", CLOCKAM; 80, 65, 1
 45 TEXT D, 195, 21, 10, 0, 2, "PM", CLOCKPM; 80, 65, 1
 TOUCH D, 65, 2, VOLMUP, N
 TOUCH D, 65, 3, VOLMDN, N
 TOUCH D, 72, 11, VOLMUP, N
 TOUCH D, 72, 12, VOLMDN, N
 50 TOUCH D, 65, 4, SETUP, N
 TOUCH D, 65, 5, SETTDN, N
 TOUCH D, 69, 3, RADONOF, N
 TOUCH D, 69, 5, HVACPWR, N
 TOUCH D, 72, 2, SWITCH, HVAC
 55 TOUCH D, 72, 5, SWITCH, RADIO
 TOUCH D, 72, 3, SWITCH, TRIP
 TOUCH D, 72, 4, SWITCH, INFO
 TOUCH D, 72, 6, SWITCH, MENU

TOUCH D, 65, 1, SWITCH, CORVETE
 TOUCH D, 72, 8, RETURN.N
 TOUCH D, 69, 2, RADONOF, N
 TOUCH D, 69, 4, HVACPWR, N

- 5 A complete description of the Graphics Design Language is shown below:
 Parameters enclosed in square brackets [] are optional.

GDI LANGUAGE

- 10 LINE -USED TO DRAW A STRAIGHT LINE BETWEEN TWO POINTS.

SYNTAX:

- 15 OPCODE PARAMETERS LINE MODE, X1, Y1, X2, Y2, COLOUR, [UNAME]

X1 - HORIZONTAL COORDINATE
 A DECIMAL VALUE IN THE RANGE OF 0-319

- 20 Y1 - VERTICAL COORDINATE
 A DECIMAL VALUE IN THE RANGE OF 0-199

X2 - HORIZONTAL COORDINATE
 A DECIMAL VALUE IN THE RANGE OF 0-319

- 25 Y2 - VERTICAL COORDINATE
 A DECIMAL VALUE IN THE RANGE OF 0-199

COLOUR - A DECIMAL VALUE IN THE RANGE OF 0-15

- 30 MODE -

- S OR D (S = DEFAULT)
 35 S - STATIC - DISPLAY ONLY
 D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

40

BOX -USED TO DRAW A BOX (OUTLINED).

- 45 SYNTAX:

OPCODE PARAMETERS BOX MODE, X1, Y1, X2, Y2, COLOUR, [UNAME]

- X1 - HORIZONTAL COORDINATE OF BOX ORIGIN
 50 A DECIMAL VALUE IN THE RANGE OF 0-39

Y1 - VERTICAL COORDINATE OF BOX ORIGIN
 A DECIMAL VALUE IN THE RANGE OF 0-199

- 55 X2 - HORIZONTAL COORDINATE OF BOX END
 A DECIMAL VALUE IN THE RANGE OF 0-319

Y2 - VERTICAL COORDINATE OF BOX END

A DECIMAL VALUE IN THE RANGE OF 0-199

COLOUR COLOUR OF BOX OUTLINE

A DECIMAL VALUE IN THE RANGE OF 0-15

5

MODE -

S OR D (S = DEFAULT)

10

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

15

CIRCLE USED TO DRAW A CIRCLE OUTLINE

SYNTAX:

20

OPCODE PARAMETERS CIRCLE MODE, X,Y,R,COLOUR,[UNAME]

X - HORIZONTAL COORDINATE OF CENTRE

A DECIMAL VALUE IN THE RANGE OF 0-319

25

Y - VERTICAL COORDINATE OF CENTRE

A DECIMAL VALUE IN THE RANGE OF 0-199

R - LENGTH OF RADIUS IN THE X DIRECTION

30

A DECIMAL VALUE IN THE RANGE OF 0-100 (IN PIXELS)

COLOUR - COLOUR OF CIRCLE OUTLINE

A DECIMAL VALUE IN THE RANGE OF 0-15

35

MODE -

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

40

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

45

FBOX -USED TO DRAW A FILLED BOX

SYNTAX:

50

OPCODE PARAMETERS FBOX MODE, X1,Y1,X2,Y2,COLOUR,[UNAME]

X1 - HORIZONTAL COORDINATE OF FBOX ORIGIN

A DECIMAL VALUE IN THE RANGE OF 0-319

55

Y1 - VERTICAL COORDINATE OF FBOX ORIGIN

A DECIMAL VALUE IN THE RANGE OF 0-199

X2 - HORIZONTAL COORDINATE OF FBOX END

A DECIMAL VALUE IN THE RANGE OF 0-319

Y2 - VERTICAL COORDINATE OF FBOX END
A DECIMAL VALUE IN THE RANGE OF 0-199

5

COLOUR - FOREGROUND COLOUR OF FILLED BOX
A DECIMAL VALUE IN THE RANGE OF 0-15

MODE -

10

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

D-DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

15

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION

20 HBAR -USED TO DRAW A BAR GRAPH IN THE HORIZONTAL DIRECTION

SYNTAX:

OPCODE PARAMETERS HBAR MODE,X1,Y1,X2,Y2,COLOUR,[UNAME]

25

X1 - HORIZONTAL COORDINATE OF HBAR ORIGIN
A DECIMAL VALUE IN THE RANGE OF 0-319

Y1 - VERTICAL COORDINATE OF HBAR ORIGIN

30 A DECIMAL VALUE IN THE RANGE OF 0-199

X2 - HORIZONTAL COORDINATE OF HBAR END
A DECIMAL VALUE IN THE RANGE OF 0-319

35 Y2 - VERTICAL COORDINATE OF HBAR END
A DECIMAL VALUE IN THE RANGE OF 0-199

COLOUR - FOREGROUND COLOUR OF HORIZONTAL BAR GRAPH
A DECIMAL VALUE IN THE RANGE OF 0-15

40

MODE - D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION

45

VBAR -USED TO DRAW A BAR GRAPH IN THE VERTICAL DIRECTION

SYNTAX:

50 OPCODE PARAMETERS VBAR MODE,X1,Y1,X2,Y2,COLOUR,[UNAME]

X1 - HORIZONTAL COORDINATE OF VBAR ORIGIN
A DECIMAL VALUE IN THE RANGE OF 0-319

55 Y1 - VERTICAL COORDINATE OF VBAR ORIGIN
A DECIMAL VALUE IN THE RANGE OF 0-199

X2 - HORIZONTAL COORDINATE OF VBAR END

A DECIMAL VALUE IN THE RANGE OF 0-319

Y2 - VERTICAL COORDINATE OF VBAR END
A DECIMAL VALUE IN THE RANGE OF 0-199

5

COLOUR-FOREGROUND COLOUR OF VERTICAL BAR GRAPH
A DECIMAL VALUE IN THE RANGE OF 0-15

MODE - D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

10

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION

TRIANGLE -USED TO DRAW A FILLED TRIANGLE

15

SYNTAX:

OPCODE PARAMETERS TRIANGLE MODE,X1,Y1,X2,Y2,X3,Y3,COLOUR,[UNAME]

20

X1 - HORIZONTAL COORDINATE OF TRIANGLE VERTEX 1
A DECIMAL VALUE IN THE RANGE OF 0-319

Y1 - VERTICAL COORDINATE OF TRIANGLE VERTEX 1
A DECIMAL VALUE IN THE RANGE OF 0-199

25

X2 - HORIZONTAL COORDINATE OF TRIANGLE VERTEX 2
A DECIMAL VALUE IN THE RANGE OF 0-319

Y2 - VERTICAL COORDINATE OF TRIANGLE VERTEX 2

30

A DECIMAL VALUE IN THE RANGE OF 0-199

X3 - HORIZONTAL COORDINATE OF TRIANGLE VERTEX 3
A DECIMAL VALUE IN THE RANGE OF 0-319

Y3 - VERTICAL COORDINATE OF TRIANGLE VERTEX 3

35

A DECIMAL VALUE IN THE RANGE OF 0-199

COLOUR - FOREGROUND COLOUR OF FILLED TRIANGLE
A DECIMAL VALUE IN THE RANGE OF 0-15

40

MODE -

S OR D (S = DEFAULT)

45

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

50

FCIRCLE -USED TO DRAW A FILLED CIRCLE

55

SYNTAX:

OPCODE PARAMETERS FCIRCLE MODE,X,Y,R,COLOUR,[UNAME]

X - HORIZONTAL COORDINATE OF CENTRE

A DECIMAL VALUE IN THE RANGE OF 0-319

5 Y - VERTICAL COORDINATE OF CENTRE

A DECIMAL VALUE IN THE RANGE OF 0-199

R - LENGTH OF RADIUS IN THE X DIRECTION

A DECIMAL VALUE IN THE RANGE OF 0-100 (IN PIXELS)

10

COLOUR - FOREGROUND COLOUR OF FILLED CIRCLE

A DECIMAL VALUE IN THE RANGE OF 0-15

MODE -

15

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

20

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION

25 TEXT -ALPHANUMERIC DATA EMBEDDED IN FRAME IMAGE

SYNTAX:

OPCODE PARAMETERS TEXT MODE,FONT,X,Y,BCOLOUR,FCOLOUR,"TEXT",[UNAME]

30

FONT - A DECIMAL VALUE SPECIFYING CHARACTER STYLE IN THE RANGE 1-3

X - HORIZONTAL COORDINATE OF FIRST CHARACTER (UPPER LEFT CORNER OF FIRST CHARACTER)

35 A DECIMAL VALUE IN THE RANGE OF 0-319

Y - VERTICAL COORDINATE OF FIRST CHARACTER (UPPER LEFT CORNER OF FIRST CHARACTER)

A DECIMAL VALUE IN THE RANGE OF 0-199

40 "TEXT" - TEXT "AS IS"

MAXIMUM NUMBER OF CHARACTERS DEPENDS ON THE FONT

45

FONT 1 = 40 CHARACTERS

FONT 2 = 40 CHARACTERS

FONT 3 = 20 CHARACTERS

MODE -

50

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

BCOLOUR-BACKGROUND COLOUR OF TEXT

55 A DECIMAL VALUE IN THE RANGE OF 0-15

FCOLOUR-FOREGROUND COLOUR OF TEXT

A DECIMAL VALUE IN THE RANGE OF 0-15

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

5

ICON -USED TO EMBED A PREDEFINED SYMBOL (48×24 PIXELS) AS A UNIT WITHIN A FRAME IMAGE

SYNTAX:

10

OPCODE PARAMETERS ICON MODE,NAME,X,Y,BCOLOUR,FCOLOUR,[UNAME]

NAME - THE MEMBER NAME OF AN EXISTING ICON WHICH RESIDES IN THE CURRENT DIRECTORY
(1 TO 7 CHARACTERS)

15

DOS FILE EXTENSION IS ALWAYS ".IMO"

X - HORIZONTAL COORDINATE FOR ICON PLACEMENT (UPPER LEFT CORNER)
A DECIMAL VALUE IN THE RANGE OF 0-319

20

Y - VERTICAL COORDINATE FOR ICON PLACEMENT (UPPER LEFT CORNER)
A DECIMAL VALUE IN THE RANGE OF 0-199

BCOLOUR - BACKGROUND COLOUR OF ICON
A DECIMAL VALUE IN THE RANGE OF 0-15

25

FCOLOUR-FOREGROUND COLOUR OF ICON
A DECIMAL VALUE IN THE RANGE OF 0-15

MODE -

30

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLER FUNCTION

35

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

40

SYMBOL -USED TO EMBED A PREDEFINED SYMBOL (16×16 PIXELS) AS A UNIT WITHIN A FRAME
IMAGE.

SYNTAX:

45

OPCODE PARAMETERS SYMBOL MODE,NAME,X,Y,BCOLOUR,FCOLOUR,[UNAME]

NAME - THE MEMBER NAME OF AN EXISITING SYMBOL WHICH RESIDES IN THE CURRENT
DIRECTORY

DOS FILE EXTENSION IS ALWAYS ".SMO"

50

X - HORIZONTAL COORDINATE FOR SYMBOL PLACEMENT (UPPER LEFT CORNER)
A DECIMAL VALUE IN THE RANGE OF 0-319

55

Y - VERTICAL COORDINATE FOR SYMBOL PLACEMENT (UPPER LEFT CORNER)
A DECIMAL VALUE IN THE RANGE OF 0-199

BCOLOUR - BACKGROUND COLOUR OF SYMBOL
A DECIMAL VALUE IN THE RANGE OF 0-15

FCOLOUR - FOREGROUND COLOUR OF SYMBOL
A DECIMAL VALUE IN THE RANGE OF 0-15

5 MODE -

S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

10 D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

15

IANIMATE USED TO CREATE SIMPLE ANIMATION USING ICONS (TIME CYCLE DISPLAY OF SERIES OF ICONS) THE CYCLE TIME CONSTANT IS SPECIFIED BY CODE IN THE CONTROLLER USING THE DELAY PARAMETER ICONS IN AN ANIMATION SEQUENCE OVERLAY EACH OTHER ANIMATION SEQUENCE OF TWO TO EIGHT ICONS IS SUPPORTED. HOWEVER, ICONS AND SYMBOLS CAN NOT
20 BE MIXED IN AN ANIMATION SEQUENCE.

SYNTAX:

OPCODE PARAMETERS

25 IANIMATE MODE,NAME[1],...,NAME[8],DELAY,X,Y,BCOLOUR, FCOLOUR,[UNAME]

NAME - THE NAME OF AN EXISTING ICON WHICH RESIDES IN THE CURRENT DIRECTORY

X - HORIZONTAL COORDINATE FOR ICON PLACEMENT (UPPER LEFT CORNER)

30 A DECIMAL VALUE IN THE RANGE OF 0-319

Y - VERTICAL COORDINATE FOR ICON PLACEMENT (UPPER LEFT CORNER)

A DECIMAL VALUE IN THE RANGE OF 0-199

35 DELAY - A VALUE FROM 0-32767 USED FOR ANIMATION DISPLAY DELAY

3 - 100 milliseconds

40 15 = 500 milliseconds

30 = 1 second

BCOLOUR - BACKGROUND COLOUR OF ANIMATION SEQUENCE
A DECIMAL VALUE IN THE RANGE OF 0-15

45

FCOLOUR - FOREGROUND COLOUR OF ANIMATION SEQUENCE
A DECIMAL VALUE IN THE RANGE OF 0-15

MODE -

50 S OR D (S = DEFAULT)

S - STATIC - DISPLAY ONLY

D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)

55

SANIMATE -USED TO CREATE SIMPLE ANIMATION USING SYMBOLS (TIME CYCLE DISPLAY OF SERIES OF SYMBOLS THE CYCLE TIME CONSTANT IS SPECIFIED BY CODE IN THE CONTROLLER USING THE DELAY PARAMETER) SYMBOLS IN AN ANIMATION SEQUENCE OVERLAY EACH OTHER. AN AMINATION SEQUENCE OF TWO TO EIGHT SYMBOLS IS SUPPORTED. HOWEVER, ICONS AND
 5 SYMBOLS CAN NOT BE MIXED IN AN ANIMATION SEQUENCE.

SYNTAX:

OPCODE PARAMETERS SANIMATE MODE,NAME[1],...,NAME[8],DELAY,X,Y,BCOLOUR, FCOLOUR,-
 10 [UNAME]

NAME - THE NAME OF AN EXISTING SYMBOL WHICH RESIDES IN THE CURRENT DIRECTORY

X - HORIZONTAL COORDINATE FOR SYMBOL PLACEMENT (UPPER LEFT CORNER)
 15 A DECIMAL VALUE IN THE RANGE OF 0-319

Y - VERTICAL COORDINATE FOR SYMBOL PLACEMENT (UPPER LEFT CORNER)
 A DECIMAL VALUE IN THE RANGE OF 0-199

20 DELAY - A VALUE FROM 0-32767 USED FOR ANIMATION DISPLAY DELAY

3 = 100 milliseconds
 25 15 = 500 milliseconds
 30 = 1 second

BCOLOUR - BACKGROUND COLOUR OF ANIMATION SEQUENCE
 A DECIMAL VALUE IN THE RANGE OF 0-15

30 FCOLOUR - FOREGROUND COLOUR OF ANIMATION SEQUENCE
 A DECIMAL VALUE IN THE RANGE OF 0-15

MODE -

35 S OR D (S = DEFAULT)
 S - STATIC - DISPLAY ONLY
 D - DYNAMIC - ASSOCIATE WITH CONTROLLER FUNCTION

UNAME - NAME OF PREDEFINED CONTROLLER FUNCTION (DYNAMIC ONLY)
 40

TOUCH -USED TO ASSIGN A CONTROLLER FUNCTION TO A TOUCH KEY

45 SYNTAX:

OPCODE PARAMETERS TOUCH MODE,X,Y,UNAME[SCREEN]

X - TOUCH GRID HORIZONTAL COORDINATE
 50 A CHARACTER IN THE RANGE OF 65 TO 69, OR 72 FOR TOUCH GRID LOCATIONS A-E OR H
 RESPECTIVELY TOUCH LOCATIONS A-E ADDRESS THE DISPLAY TOUCH GRIDS TOUCH LOCATION H
 ADDRESSES A HARD SWITCH

Y - TOUCH GRID VERTICAL COORDINATE
 55 A DECIMAL VALUE IN THE RANGE OF 1-5 IF X IN RANGE (65-69)
 OR
 A DECIMAL VALUE IN THE RANGE OF 1-23 IF X = 72 (H)

UNAME - THE NAME OF A PREDEFINED CONTROLLER FUNCTION (PREDEFINED FUNCTIONS ARE IMPLEMENTATION DEPENDENT)

SCREEN - FRAME NAME IF UNAME = "SWITCH"

5

MODE - D - A TOUCH STATEMENT IS DYNAMIC BY DEFINITION

DATE -DATE FOR FRAME CREATION

10

SYNTAX:

OPCODE PARAMETERS DATE - DATE-STRING

15

DATE-STRING - THE DATE OF FRAME CREATION (OR LAST UPDATE TO THE SCREEN)

BCOLOUR -BACKGROUND COLOUR OF FRAME

20

SYNTAX:

OPCODE PARAMETERS BCOLOUR - COLOUR

COLOUR - BACKGROUND COLOUR OF FRAME

25

A DECIMAL VALUE IN THE RANGE OF 0-15

ECOUNT -ELEMENT COUNT

30

SYNTAX:

OPCODE PARAMETERS ECOUNT - NUMBER

NUMBER - NUMBER OF ELEMENTS IN FRAME

35

A DECIMAL VALUE IN THE RANGE OF 1 - N, WHERE N IS THE TOTAL NUMBER OF GRAPHIC AND TOUCH ELEMENTS IN A FRAME

REF -REFERENCE NUMBER

40

SYNTAX:

OPCODE PARAMETERS REF - REFERENCE-STRING

45

REFERENCE-STRING-A STRING OF CHARACTERS TO DESCRIBE THE FRAME (USUALLY A MANUAL PAGE NUMBER)

50

AUTHOR -AUTHOR'S IDENTIFICATION

SYNTAX:

OPCODE PARAMETERS AUTHOR - AUTHOR-ID

55

AUTHOR ID - AUTHOR'S IDENTIFICATION
A STRING OF CHARACTERS

FLAGS -FRAME MODE FLAGS

SYNTAX:

5 OPCODE PARAMETERS FLAGS - S-FLAGS

S-FLAGS - FRAME MODE FLAGS

10 0 = FRAME CAN BE DISPLAYED ANYTIME
 1 = FRAME CANNOT BE DISPLAYED WHILE THE VEHICLE IS IN MOTION
 2 = FRAME CANNOT BE DISPLAYED WHILE IN IGNITION ACCESSORY MODES
 3 = FRAME CANNOT BE DISPLAYED WHILE THE VEHICLE IS IN MOTION OR WHILE IN
 IGNITION ACCESSORY MODES

15

STYPE -FRAME TYPE

SYNTAX:

20

OPCODE PARAMETERS
 STYPE - S-TYPE

S-TYPE - FRAME TYPE

25

A DECIMAL NUMBER IN THE RANGE 0-9

0 = NORMAL
 1 = LOGO
 2 = SUMMARY
 3 = MAIN RADIO
 4 = CHECK SERVICE
 5 = CHECK EVENT
 35 6 = IMMEDIATE MESSAGE
 7 = DIAGNOSTIC GRAPHICS
 8 = DIAGNOSTIC TEXT
 9 = SYSTEMS OK

After a frame is designed and is saved as a GDL text file, it must be post-processed and assembled to
 40 create the necessary tables and data which identify and describe the frame's Graphical elements, Poll
 functions, Animation sequences, and Touch functions. The Graphics Development Instrument System, Post-
 Processor program translates the GDL text file into a frame assembly file and adds the necessary macro
 function calls to allow the macro assembler to create the appropriate data table which allow the Control
 program to properly display and update the frame's graphical elements. Custom Post-Processor assembler
 45 macros are required for the proper assembly of a frame's GDL file. In addition to the macros, several other
 files which define the Dynamic Element Names, Touch functions, and Animation functions which are
 supported by the Control program are required by the Assembler. These files may be changed depending
 on the functions supported by various models of the Controller. The frame assembly file is then assembled
 with the Macro Assembler to create a frame object file. Frame object files are combined into a frame set by
 50 the DOS Link program to create a frame set EXE file. This file is then converted to a binary file to
 programmed into an EPROM for use in a Controller.

Frame Data Structure

55

The Table structure of the Frame data consists of a global header, a frame header for each frame in the
 frame set, and Graphic display data for each frame in the frame set. The Table structure of this binary file is
 shown in Fig. 20. Static graphic elements are present in each frame's data and are displayed when the

frame is first displayed. Also, depending upon the design of each frame, additional data tables may be present in a particular frame's data to describe Dynamic Graphical Elements as well as Touch, Poll, and Animation Functions which were specified via the GDIS when each frame was designed. These additional tables are used by the Control program to identify and manipulate the Dynamic Graphic Elements, perform
 5 Touch functions including switching to new frames, performing Poll functions, and performing Animation functions.

The typical display frame shown in Fig. 1 is composed of frame elements. There are two basic types of frame elements, Static elements and Dynamic elements. Static elements are shown in Fig. 1 as items 11, 12, 13, 14 and 15. These elements are lines and boxes which will be drawn on the Display frame and will
 10 never be changed until a new frame is displayed. Other elements in the frame which are Static are elements 16 and 17. Element 16 is a static text element VOL representing the volume up and down for the radio function. Item 17 is a Static symbol representing a down arrow for the set temperature controls. Also shown is Item 22, a Dynamic text element for the time of day clock. Dynamic Icons 20 and 21 represent power switches for the radio and climate control system respectively. Other Dynamic elements include
 15 Items 23 and 24 for the Radio station display and inside Set Temperature display. The display frame in Fig. 1 is typical of that generated using the GDIS described here and displayed on the vehicle Display system. Touch functions associated with the Display are also assigned using the GDIS.

Fig. 2 represents the interior of a typical automobile. Illustrated in Fig. 2 is a dashboard 30, an instrument panel 13, a data processing display and computer unit 32, data processing keyboard 33, CRT
 20 information centre 36, steering wheel mounted controls 35, console 31, with compact disk player 40, tape cassette player 42, push button transmission selector 37, and cellular radio telephone 41. Display frames such as that depicted in Fig. 1 would typically be displayed on the CRT Information Centre 36.

Fig. 3 is a block diagram of the Vehicle Display System hardware showing a Display Controller 34, Display 36a, Touch Controller 36b, and Switch Assemblies 36c. Also shown are various interconnecting
 25 signal lines. A system of this nature is described in the above mentioned copending patent application.

Fig. 4 is a high level block diagram of the Display Controller hardware. Shown are an 80C88 processor system block 50, EEROM block 51, RAM block 52, ROM block 56, and Watch Dog Timer block 55. An Interprocessor Data Link based on an 82C55 chip is shown in block 57. An 80C31 Microprocessor system block 60 is interconnected with ROM block 61, RAM 62, and Data Links block 63, via bus 71. Bus 70
 30 interconnects the 80C88 system block 50 with the Video circuitry block 65, Touch Switch circuitry block 67 and Discrete input circuitry block 68. Also shown is Power Supply circuitry block 53.

Fig. 5 is a flow chart of the Frame development process. Blocks 80 and 81 illustrate Screen, (Frame), Symbol, and Icon designer inputs from the GDIS keyboard. Inputs from the keyboard are accepted by blocks 82 and 83, the GDIS Icon and Symbol design program and GDIS Screen Generator programs
 35 respectively. The output of the GDIS Icon and Symbol design program, block 82, creates Icon source files, block 91, and/or Symbols source files, block 92. The output of the GDIS Screen Generator program, block 83, generates frame definition files, block 84. Block 85 depicts the GDIS Post-Processor program which accepts Screen Definition files 84, Touch Function Definition files 87, Poll Functions and Dynamic Element Definitions file 88, and Animation Function Definitions file 89 as input. Its output is Screen Source files,
 40 block 86, which are assembled along with the Symbol Source files, block 92, and Icon Source files block 91, using the GDIS Post-Processor Macro file block 90, by the the Macro Assembler program, block 93, which creates Screen Object files, block 94, which are then linked with other Screen Object files in a particular vehicle Screen,(Frame), set by Object Module Link program block 95, to create the relocatable vehicle Screen Set Data file, block 96. This file may then be relocated by the GDIS Screen Data relocation
 45 program to create an EPROM Image Screen Data file which is then used to create the EPROM chip which will reside in the Controller hardware.

The following tabulates various programs utilised in the development process.

50

55

	Program Name	Operator	Action
5	GDIS Utility	Screen Designer	Interactively Design Screens
	GDIS Post-Processor	Screen Designer	Create Control Tables
10	Macro Assembler	Screen Designer	Create Screen Object Data File
15	DOS Link	Screen Integrator	Combine Screens into Screen Set
20	EXE2BIN	System Integrator	Create Screen Set EPROM image
25	EPROM Programmer	System Integrator	Program Screen Set into EPROM

Fig. 6 is a drawing of a typical GDIS workstation 100. Components of the workstation include the keyboard 103 for the GDIS menu display, the CRT monitor 102, system unit 104, diskette drive 105, fixed disk 106, vehicle display 107 and desk 110.

Fig. 7 is a listing of the GDIS workstation hardware. Shown are a personal computer 104 with hard file, keyboard 103, and dual displays 102 and 107. Also listed but not shown are the Printer and Host communications adapter.

The Graphic Display Instrument System (GDIS) hardware is further illustrated in Fig. 8 in block diagram form. In this case, it includes a PC XT/AT (Personal Computer products of International Business Machines Corporation) based unit with a modified expansion chassis. The PC XT/AT 104 contains an Enhanced Graphics Adapter (EGA) card 402 which provides video signals to a video adapter card 403 (PL100) in the expansion chassis 406 through a coaxial cable 407 (PF301). The video adapter card in the expansion chassis modifies the video signal to support a monochrome monitor 410 and a colour monitor 411. Power to each monitor is provided by a power card (PL101) designed specifically for the GDIS.

The power card 415 detailed in Fig. 9 provides an interface for the power supply 416 in the expansion chassis and for the vertical sync signal to one monitor (monochrome shown connected) via cable assembly 417 (PF300). The power supply +12V and ground connect to CONN1 420 and exit through the 25 pin D shell connector 421 (CONN2). The +12V is fused at 425 to limit current to 4 amperes. The vertical sync signal enters the card via edge connector 430 from the expansion chassis mother board and exits at connector 421 CONN2. A second power card is required if two monitors are used.

The video card 403 is detailed in Fig. 10 and supports one monochrome monitor with analog interface and a second monitor (colour) with 16 colour digital interface (R, G, B, and I), using composite sync or separate horizontal and vertical sync.

The input video signal to the card appears at connector 435 CONN1. The video signals to the colour display are buffered without inversion through jumper connector 436 CONN2 (shown in Fig. 10). Figure 8 illustrates jumper 440 J2 connecting the colour video signals to the colour monitor. Fig. 10 shows the combining of the input horizontal and vertical sync at block 445 (U003D-an Exclusive Or Gate) to form the composite sync signal required by some colour monitors. The horizontal and vertical sync input signals are also buffered by block 446 and 447 (U003A and U003B) for colour monitors requiring separate vertical and horizontal sync.

The output signals for a monochrome monitor appear at jumper connector 450 CONN3. The same type of drive circuit is used by the video and horizontal sync. The video signal for mono is buffered by block 451

U1 and inverted by block 452 (U002C). Block 452 drives transistor Q002 which is used to drive the G (green) video signal as a video input for the monochrome monitor. Resistor R007 provides current limiting and resistor R008 is used for pull up. The vertical sync signal is buffered twice by blocks 460 and 461 and is output in open collector configuration to connector 430 B27, the expansion chassis mother board and to
 5 connector 450 CONN3 without inversion.

Summary of GDIS Hardware Features:

- 10 (1) Use of the actual target monitor that will be used in the production vehicle. Aspect ratio problems are eliminated.
- (2) Flexibility: A monochrome and colour monitor can be used independently or concurrently when jumpers J1 and J2 are installed on the video card as required and when a power card is installed for each monitor.
- 15 (3) The video card supports both analog and digital video monitor interfaces.
- (4) Support is provided for colour monitors with either composite or separate sync.
- (5) The video interface will support various screen sizes (5V or 13V) and resolutions.
- (6) With minor modification the video interface will support EL flat panels and colour monitors with 64 colour capability.
- 20 Fig. 11 is a listing of the GDIS workstation software. Software required includes the Vehicle Screen Generator program, the Post-Processor program and Host Communications support software.
- Fig. 12 represents the functions of the Vehicle Screen Frame Generator program. It allows the creation, editing and saving of frame designs which may include Graphic Elements, Icons and Symbols, Touch Screen (Panel) functions, and Animation.
- 25 Fig. 13 represents typical Graphic Elements which may be used while designing frames with the GDIS. Graphic Elements include boxes 116, lines 115, filled boxes 117, circles 118, filled circles 119, triangles 120, filled triangles 121, horizontal bar graphs 123 and vertical bar graphs 122.
- Fig. 14 illustrates the action of the Post-Processor program which creates screen frame source files which may be combined to create a vehicle screen (frame) set consisting of many different frames. Shown
 30 are a typical Icon fan 132, a Clock frame 133, and a vehicle drawing frame 131 which are combined to create vehicle Screen (Frame) Set 130.
- Fig. 15 represents the use of the Vehicle Screen (Frame) Set by the Display Controller 34, to create Screen (Frame) images, Automobile functions and Animation, in the vehicle.
- Fig. 16 depicts the Host communication capability of the GDIS. Shown is a Host system 142,
 35 interconnected with a GDIS Gateway system that includes personal computer 104a, monitor 102a, and keyboard 103a. Also shown is a Remote GDI system with personal computer 104, monitor 102 and keyboard 103. The Host 142 and GDIS Gateway system are directly connected. The telephones 140 and 141 on the GDIS Gateway system and Remote GDI system respectively allow communication between the two GDI systems.
- 40 Fig. 17 depicts the typical GDIS networking system with Network 150. The Network consists of a Host Processor 142 with direct attached GDIS workstations 100a-100c and Remote GDIS workstation 100.
- Fig. 18 is a flow chart of the GDIS Control program. The program is automatically started at block 160, when the GDI system is powered on. A logo screen (frame) is displayed and the vehicle display is initialised, block 161. Next the GDIS main menu is displayed, block 162. The user then enters a menu
 45 selection on the Keyboard block 163. If Option 1 is selected, block 165, then the Screen Edit menu is displayed, block 170. The user will then select from the Screen Edit menu block 171. The command is then processed by block 172, and the display of the Edit menu is then redrawn, block 170. This action repeats until block 171 menu item Save is selected. The Screen Definition file is then saved on the hard disk block 173 and the GDIS main menu is again displayed, block 162. If main menu Option 2 is selected, block 166,
 50 then an existing Screen Definition file is read into work space, block 174, and the Screen Edit menu is displayed, block 170, to allow the screen (frame) to be modified. If Option 3 is selected from the main GDIS menu, block 167, an existing Screen Definition file is deleted, block 175, and the main menu is again displayed. If Option 4 of the main menu is selected, block 168, then the Icon and Symbol Editor menu is displayed, block 176. The user then inputs Icon and Symbol editor commands, block 177, and they are
 55 processed one at a time by, block 178. The Icon Symbol Editor sub-menu is again displayed. This process repeats until the Exit command is entered. At this point the GDIS main menu is again displayed. If Option 5 is selected from the main menu block 169, then the Host Communication Sub-Menu is displayed, block 180. The user then inputs the Host Communication selection from the Keyboard, block 181, which is then

performed by Host Communication sub-routines block 180. This process repeats until Exit is selected on the Host Communication menu. The GDIS menu is then displayed.

Fig. 19 is a flow chart of the GDIS Post-Processor program. The Post-Processor program is entered with a specified Screen Definition file name to be processed, block 190. The Post-Processor program then displays a copyright notice and opens the Screen Definition file which is to be processed, block 191. Next, the Screen Source file which is to be created is opened, block 192. The screen (frame) header is then written to the output file, block 193. Two-hundred fifty-six bytes of the Screen Definition file are then read into a Buffer, block 194. Each element of the Screen Definition file is then processes one at a time. If the screen (frame) element is static the element needs no conversion, block 195. If the screen (frame) element is dynamic then the dynamic element name must be set to its defined value, block 196. The element, either static or dynamic, is then written to output file, block 197. A check is then made if all of the elements in the Buffer have been processed, block 198. If they have not, then control passes back to process the next element, block 195. If all Buffer elements have been processed then control passes to block 199 where a test is made to determine if the complete screen file has been processed. If the screen file has been completely processes, the screen file tables are then written to the output file block 200. The output file is then closed, block 201, and the program exited, block 202.

Fig. 20 illustrates the Screen Data Structure produced by the GDIS Post-Processor program. It consists of a Global Header block 210, a Screen header for each screen block 211, a Switch Screen Table block 212, a Statics Pointer Table block 214, Static Elements block 215, Icon and Symbol Data block 216, Dynamics Pointer Table block 217, Dynamic Elements block 218, Poll Table block 219, Touch Table block 220, Animation Pointer Table block 221, Animation Element 1 block 222, and Animation Element N block 223. These tables fully define all elements used to compose the Display screen (frame). These tables are used by the Controller software to display and control the screen (frame) image.

Fig. 21 is a flow chart of the Touch Function Handler of the Controller software. It is called from the Supervisor program with a Touch Key number, block 230. The current Screen Header is then pointed to, block 231. If in the Initialise mode, block 232, the Initialise key number is set to 1 and Touch key number is also set to 1, block 233. If not in Initialise mode then the number passed from the Supervisor is left untouched. Next the Key number is inspected to be in the range from 1 to 48, block 234. If the number is out of the valid range then an error is returned to the Supervisor, block 244, otherwise the pointer to the current screen (frame) Touch Function Table is obtained, block 235, from the screen's (frame's) header. Block 236 connector is then followed to block 237 where a determination is made if any Touch functions are defined on the screen (frame), if no functions are defined then the Supervisor is returned to with no error. If functions are defined, the particular Touch function number for the specified key is obtained from the Touch Function Table, block 238. Next, it is determined if the function is a Switch Screen function, block 239. If not a Switch Screen function, a determination is made if the Touch Function is valid, block 240. If the function is valid it is performed, block 241, and if not in the Initialise mode, block 242, the Supervisor is returned to with no error, block 250. If the Touch Function is invalid, return is passed to the Supervisor with an error indicated without performing the Touch function. If the valid Touch function was selected in the Initialise mode then the Initialise Key number is incremented and the Touch key number is set to that value. Control then passes to block 234. If the Touch function selected is a Switch Screen function block 239, control is passed to block 246. If Initialise mode in active then return is passed to the Supervisor with no error, otherwise the Switch Screen Table pointer is obtained from the current Screen Header, block 247, and the pointer to the new screen (frame) to be switched to is obtained, block 248. The new Screen Loader routine is then executed, block 249, and the Supervisor is then returned to with no error at block 250.

Fig. 22 is a flow chart of the Poll Function Handler. The Poll Function Handler is entered from the Supervisor block 255. First the Poll Function routine table is pointed to, block 256. The current Poll Table Index is obtained, block 257, and the Poll Table Index is updated for Next Time, block 258. The current Screen Header is then pointed to, block 259, the pointer to the Poll Table is then obtained from the Screen Header, block 260, and a determination is made if any Poll functions are defined on the current screen (frame), block 261. If none are defined then the Poll Table Index is reset to zero, block 262, and the Supervisor is returned to with no error, block 268. If Poll Functions have been defined on the current screen (frame) then the Poll Function to be performed is obtained using the Poll Index, block 263. If this is the end of the Poll Table, block 264, then the Poll Table Index is also reset to zero, block 262, otherwise a determination is made if the Poll Function is valid, block 265. If not valid the Supervisor is returned to with an error indicated, block 266. If the Poll Function is valid, the Poll Function is performed, block 267, and return is passed to the Supervisor with no error, block 268.

Fig. 23 is a flow chart of the Animation Function Handler. The Animation Function Handler is entered from the Supervisor block 270. The current Screen header is pointed to, block 271, the current screen's

Animation Table Pointer is obtained from the Screen Header, block 272, and a determination is made if any animation is on the current screen, block 273. If no Animation is present then the Supervisor is returned to with no error, block 274, otherwise the first frame delay counter is pointed to, block 275. The current Animation Function type is obtained, block 276, and a determination is made if this is the end of the Animation Table, block 277. If the end of the table is encountered return to the Supervisor with no error occurs, block 274, otherwise a determination is made if the Animation Function is valid, block 278. If the Animation Function is invalid, return to the Supervisor with an error is performed, block 279, otherwise the Animation Function is valid and the function is performed, block 280. The next Animation Table entry and next frame/delay counter are then pointed to, block 281. If more than eight Animation sequences exist on a single screen, block 282, then the remaining sequences are ignored, block 282, otherwise control is passed back to block 276 to get the next Animation Function Type.

Fig 24 is a flow chart of the Screen Loader Routine. It is entered with a pointer to the Screen header of the new Screen which is to be displayed. If the vehicle is in motion and the new screen (frame) is not allowed while the vehicle is in motion, then return occurs without displaying the new screen (frame), blocks 286, 287 and 302. Or, if the current operating mode is Accessory Mode and the screen (frame) cannot be displayed in Accessory Mode, then return occurs without displaying the new screen (frame) also, blocks 288, 289 and 302. Otherwise the previous screen pointer is set equal to the current screen pointer, block 290, the new screen header is pointed to, block 291, the new screen background colour is obtained, block 292, the background Buffer is selected for drawing, block 293, the background Buffer is cleared to the background Colour, block 294, the Static screen elements are then drawn block 295, the Touch Functions are then initialised, block 296, Poll Functions are initialised, block 297, and Animation Functions are initialised, block 298. Next, English or Metric Units are displayed depending upon the current mode, block 299. The new screen is then displayed from the background Buffer, block 300, and return is executed with the new screen now displayed, block 31.

25

Claims

1. A system for generating frames of information for displaying on a display system of the type including a display unit (36), a display controller (34), and a control program for operating said display controller, comprising

a data processor (93) including associated storage and display means, and
a generating program for said data processor for selectively generating frame data, and editing and storing said data, said data being representative of graphics elements, symbols, icons and animation required for each frame to be generated,

characterised in that

said generating program operates independently of said control program, whereby said frame data can be generated without altering said control program.

2. A system as claimed in claim 1 characterised in that said generating program comprises an auxiliary program for selectively combining said generated from data so as to create at least one set of frames for a particular display system.

3. A system as claimed in claim 1 or claim 2 characterised in that said data processor comprises a display monitor similar to the display unit of said display system for displaying frame data while it is being generated.

4. A system as claimed in any one of the preceding claims characterised in that said data processing system comprises a host processor for performing selected frame generating functions, a remote processor for performing other frame generating functions, and a communication link between said host processor and said remote processor.

55

Neu eingereicht / Newly filed
Nouvellement déposé

FIG. 1

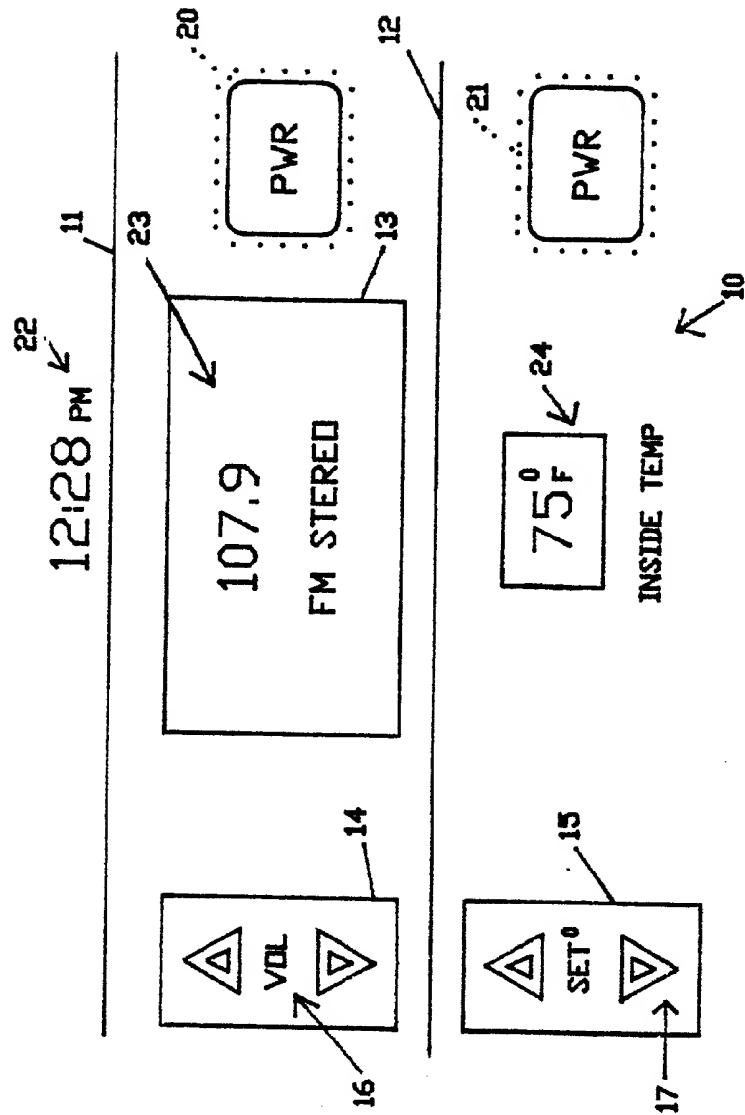


FIG. 2

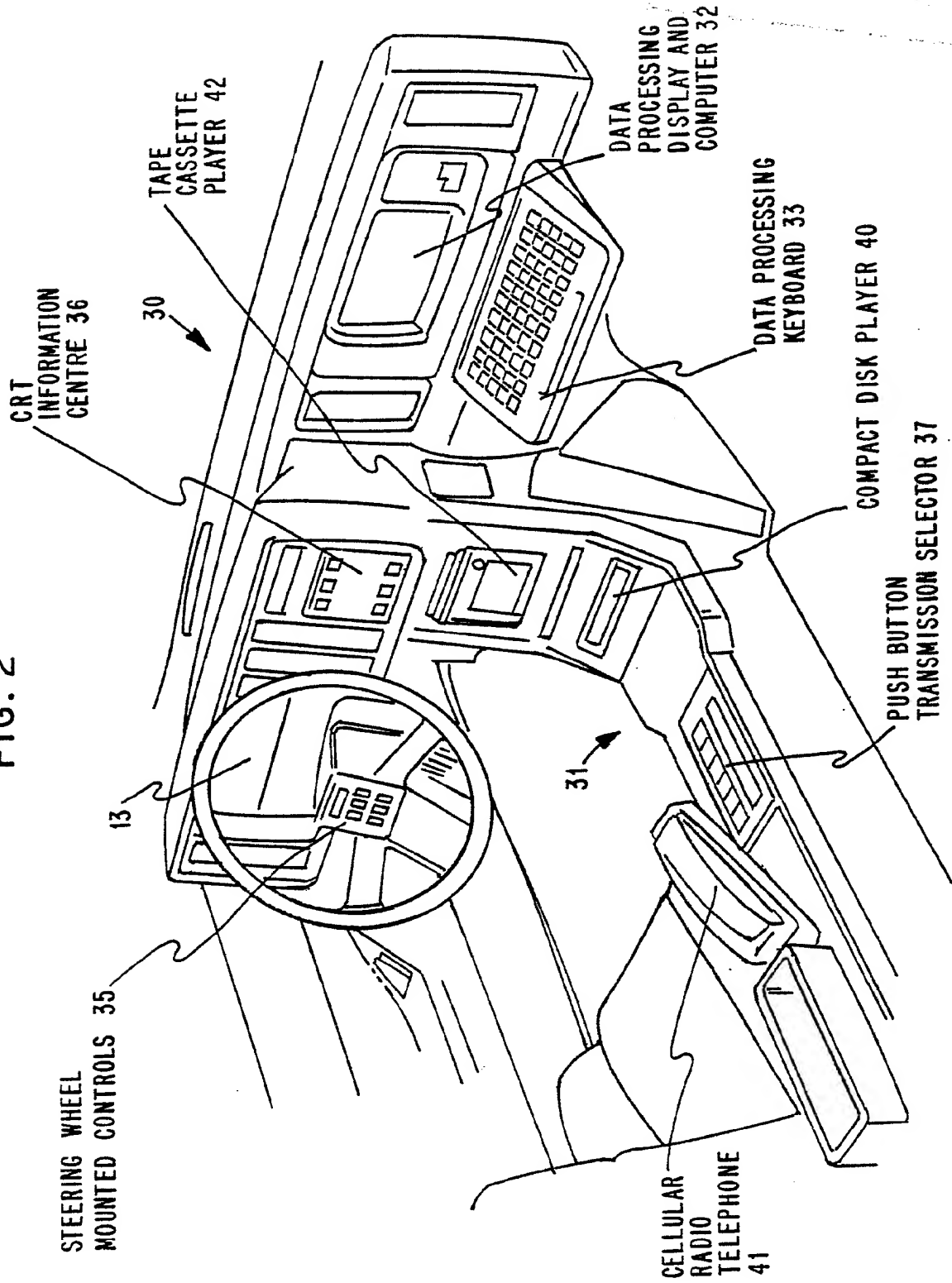


FIG. 3

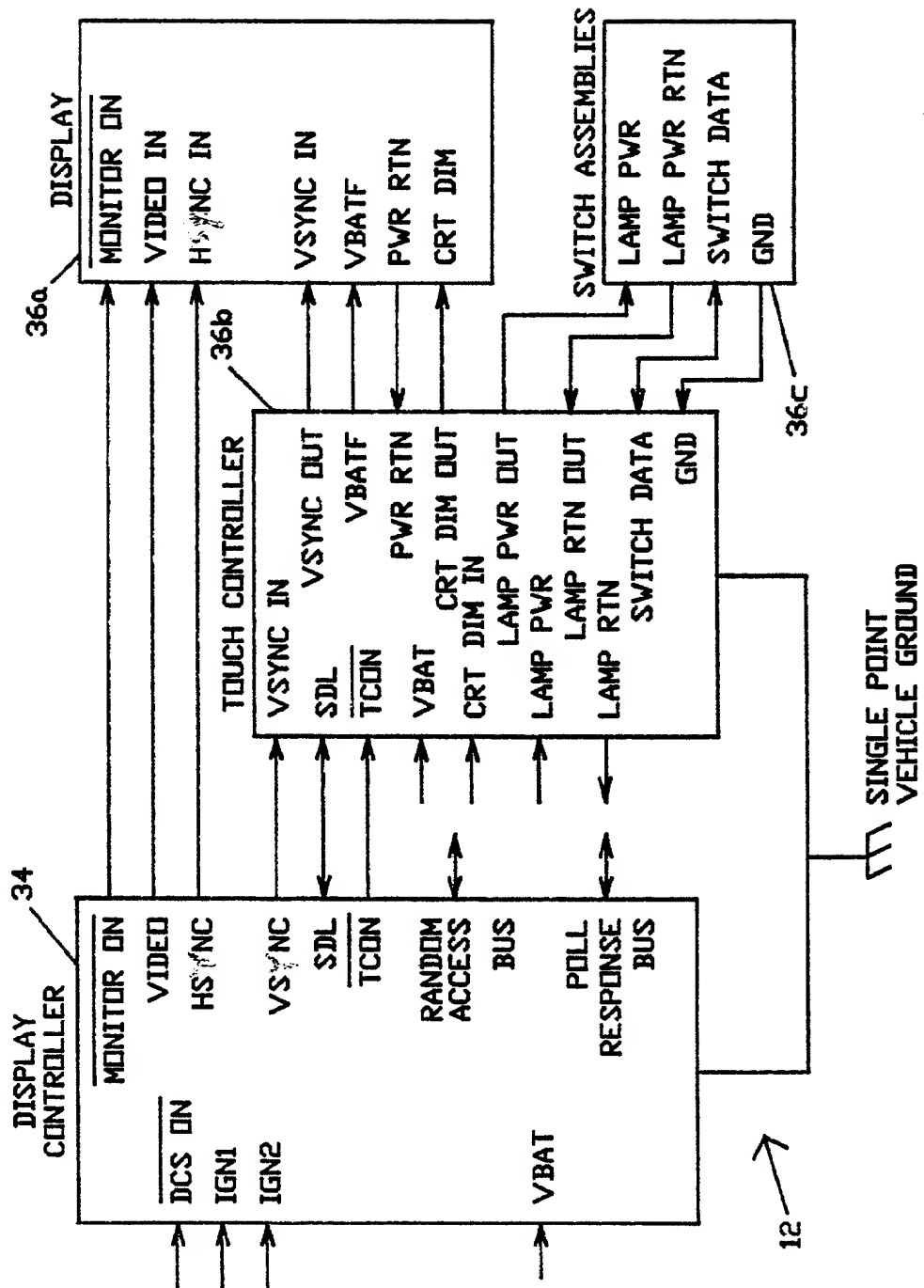
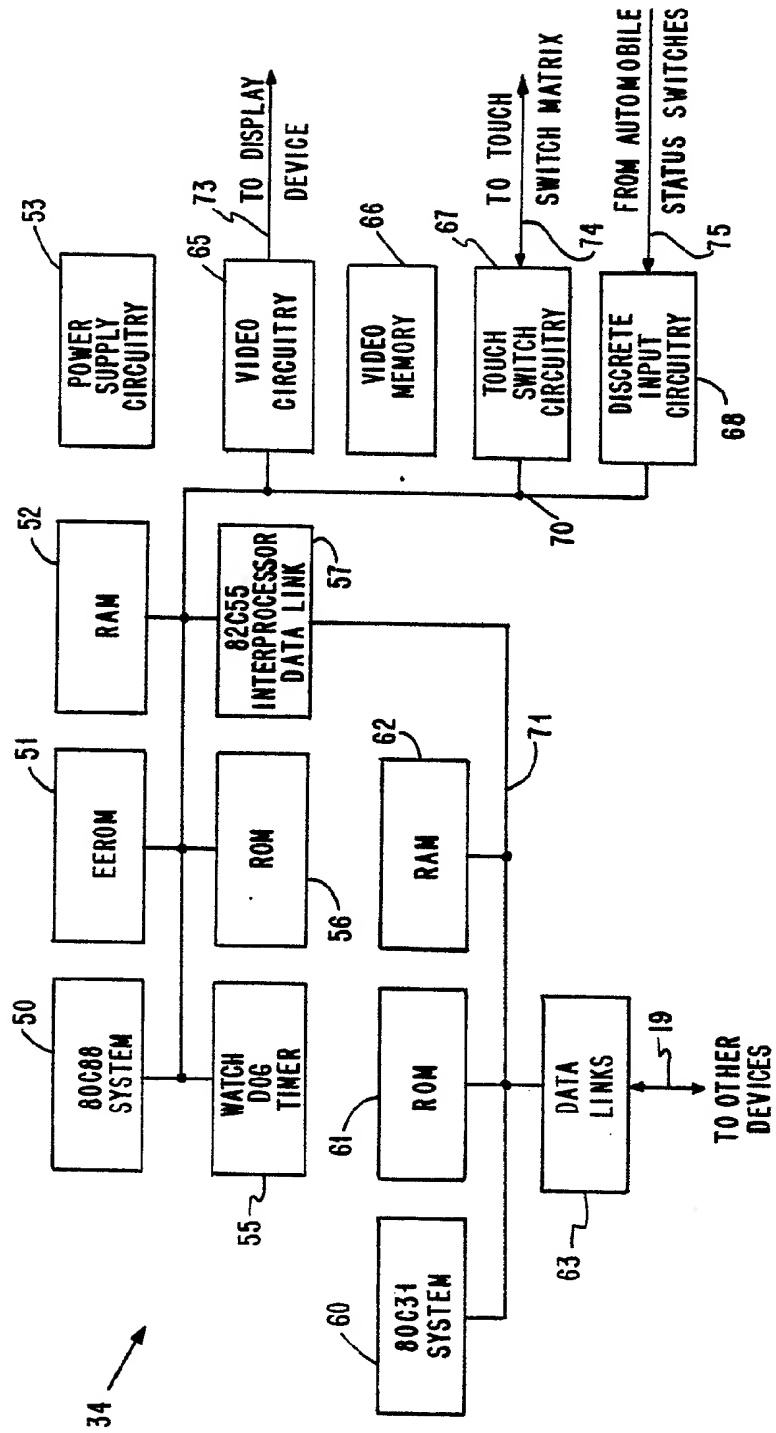


FIG. 4

DISPLAY CONTROLLER - VERSION 1



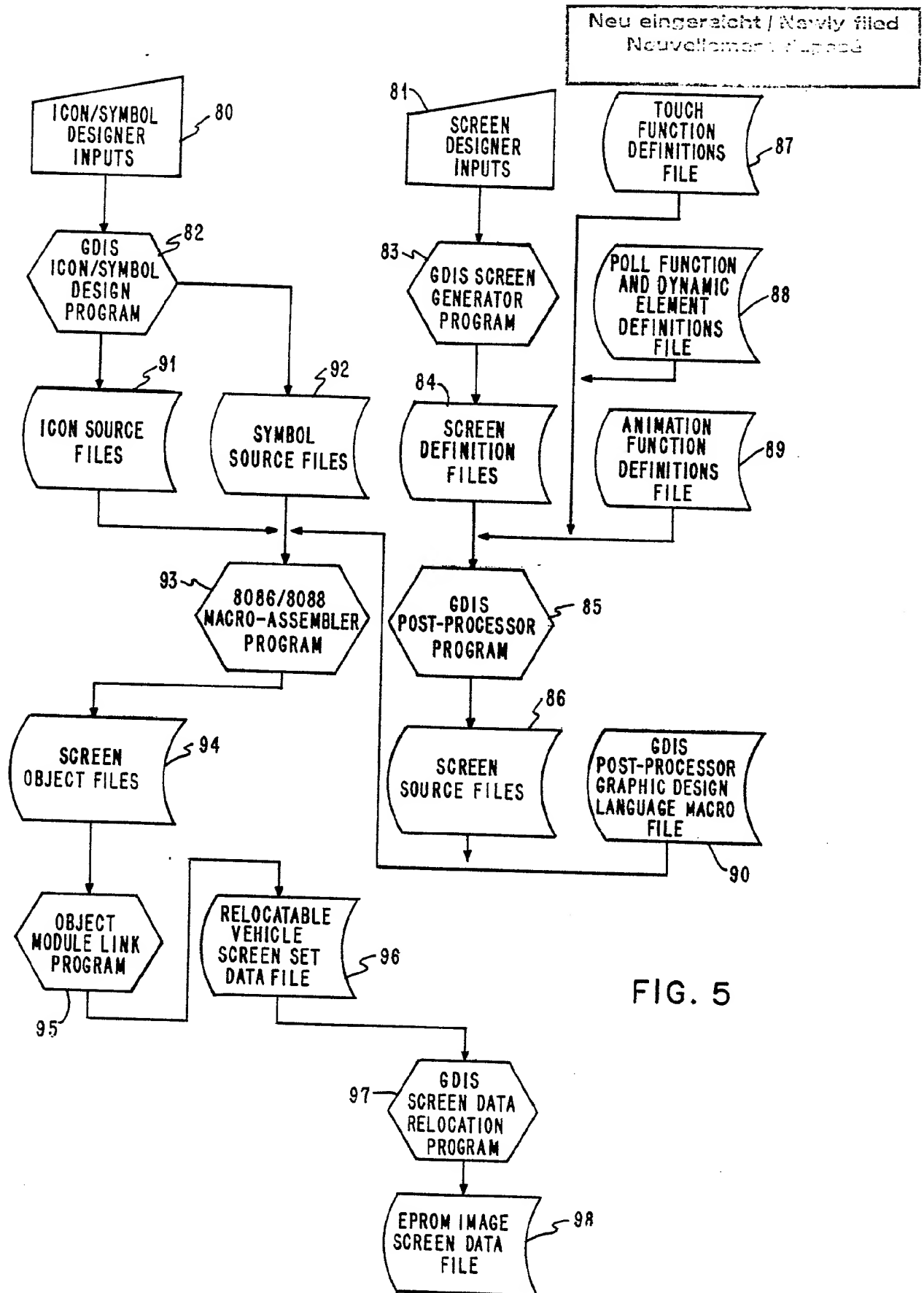


FIG. 5

FIG. 6

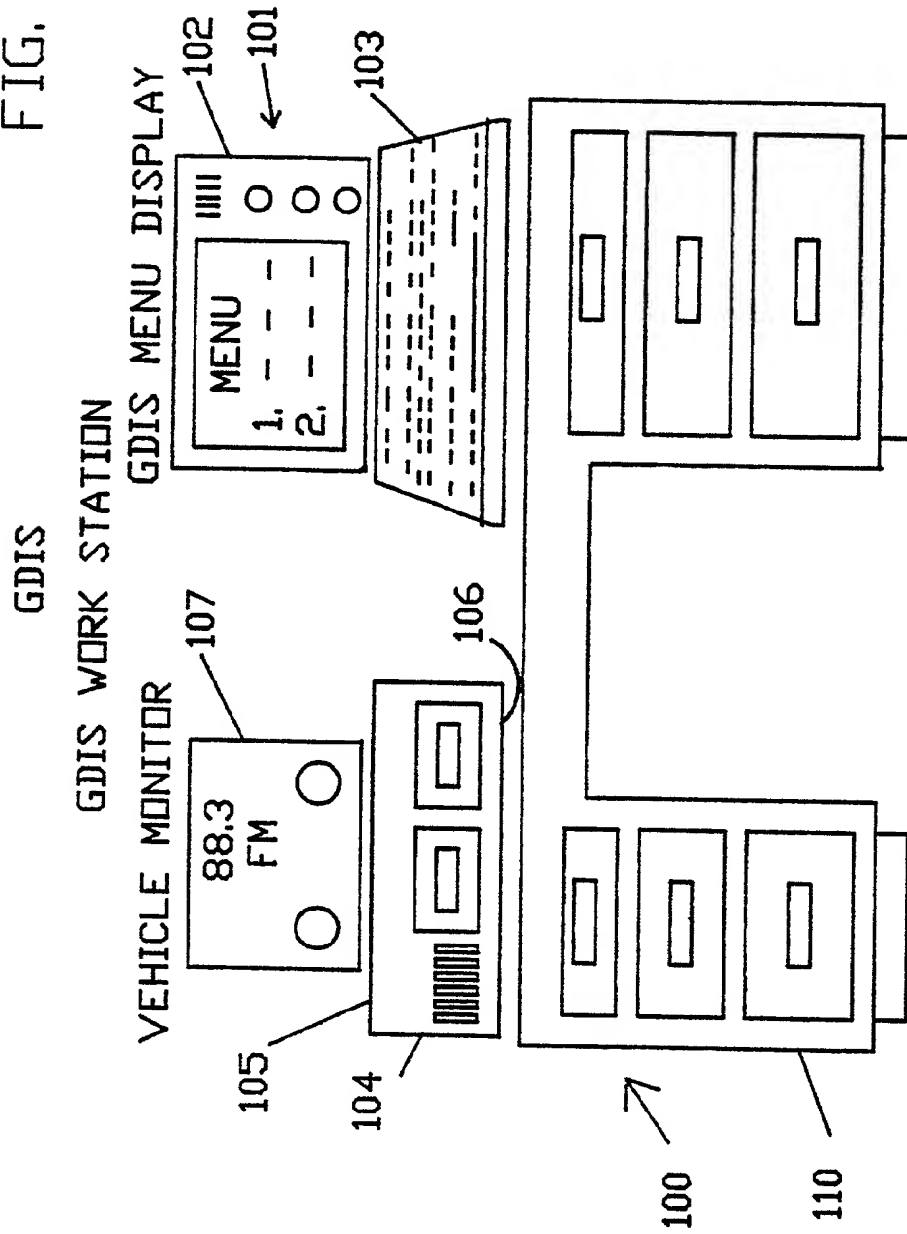


FIG. 7

GDIS

GDIS WORK STATION

HARDWARE

* PERSONAL COMPUTER WITH
HARD FILE

* 256 KB MEMORY

* DUAL DISPLAYS

* PRINTER

* HOST COMMUNICATIONS ADAPTER

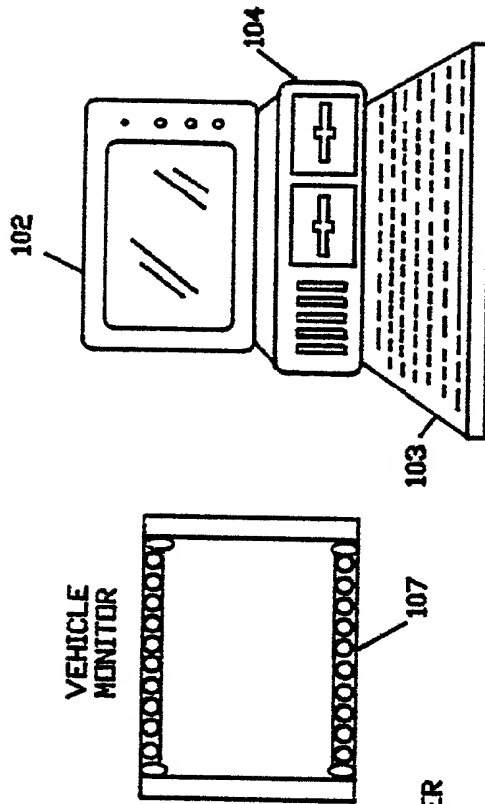
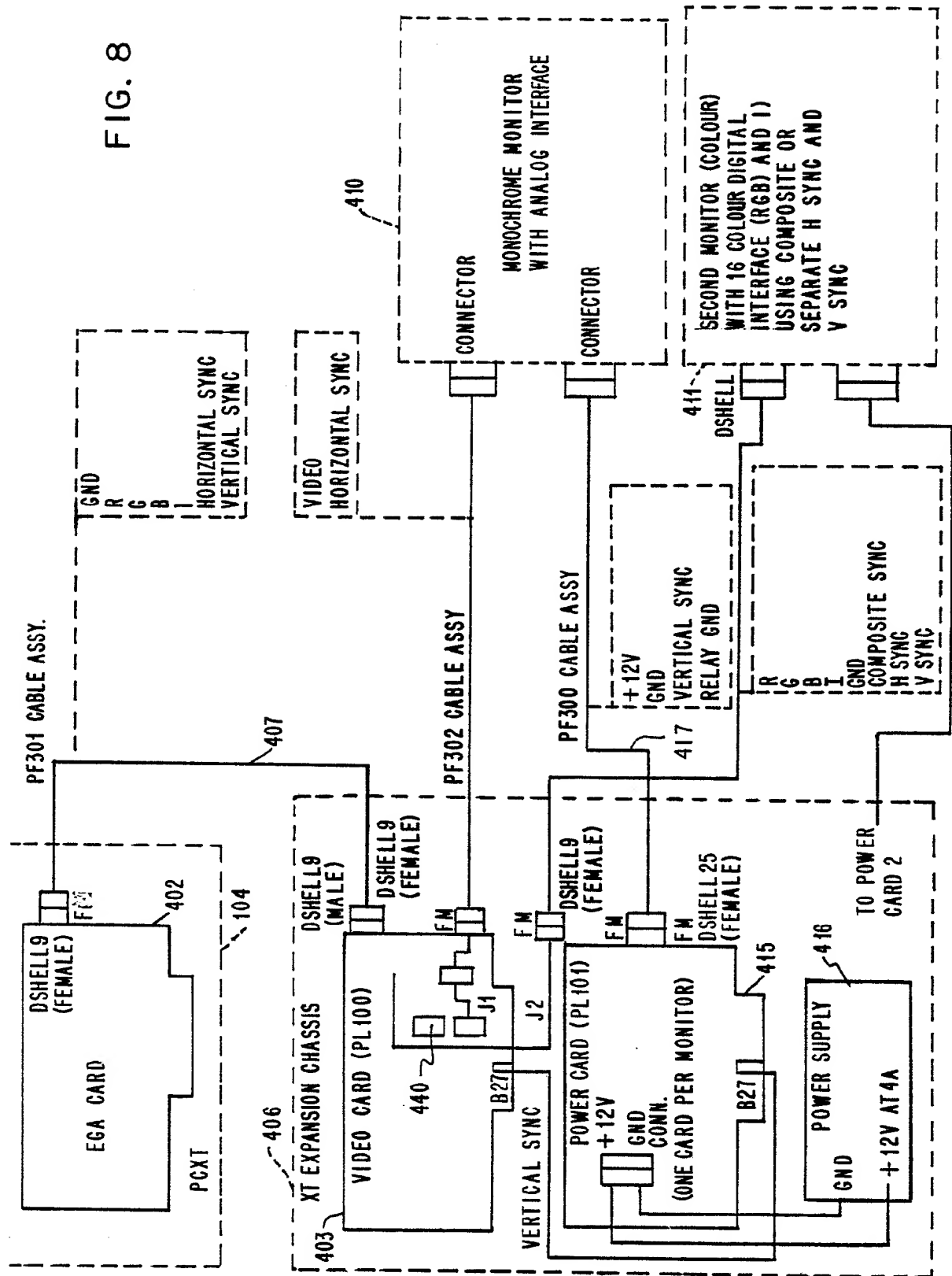


FIG. 8



Neu eingereicht / Newly filed
Nouvellement déposé

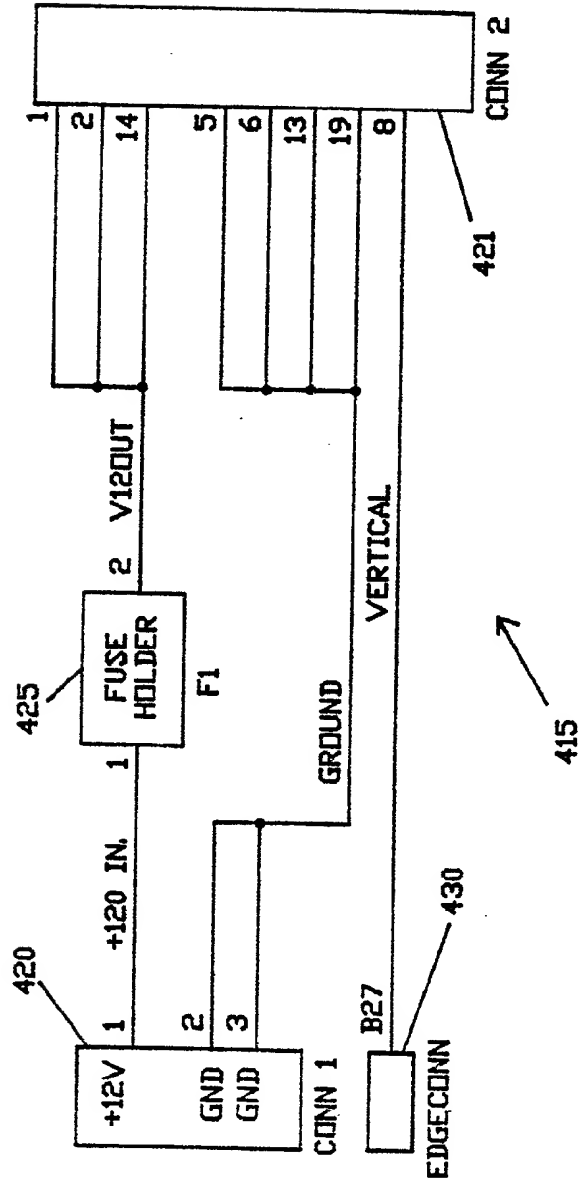
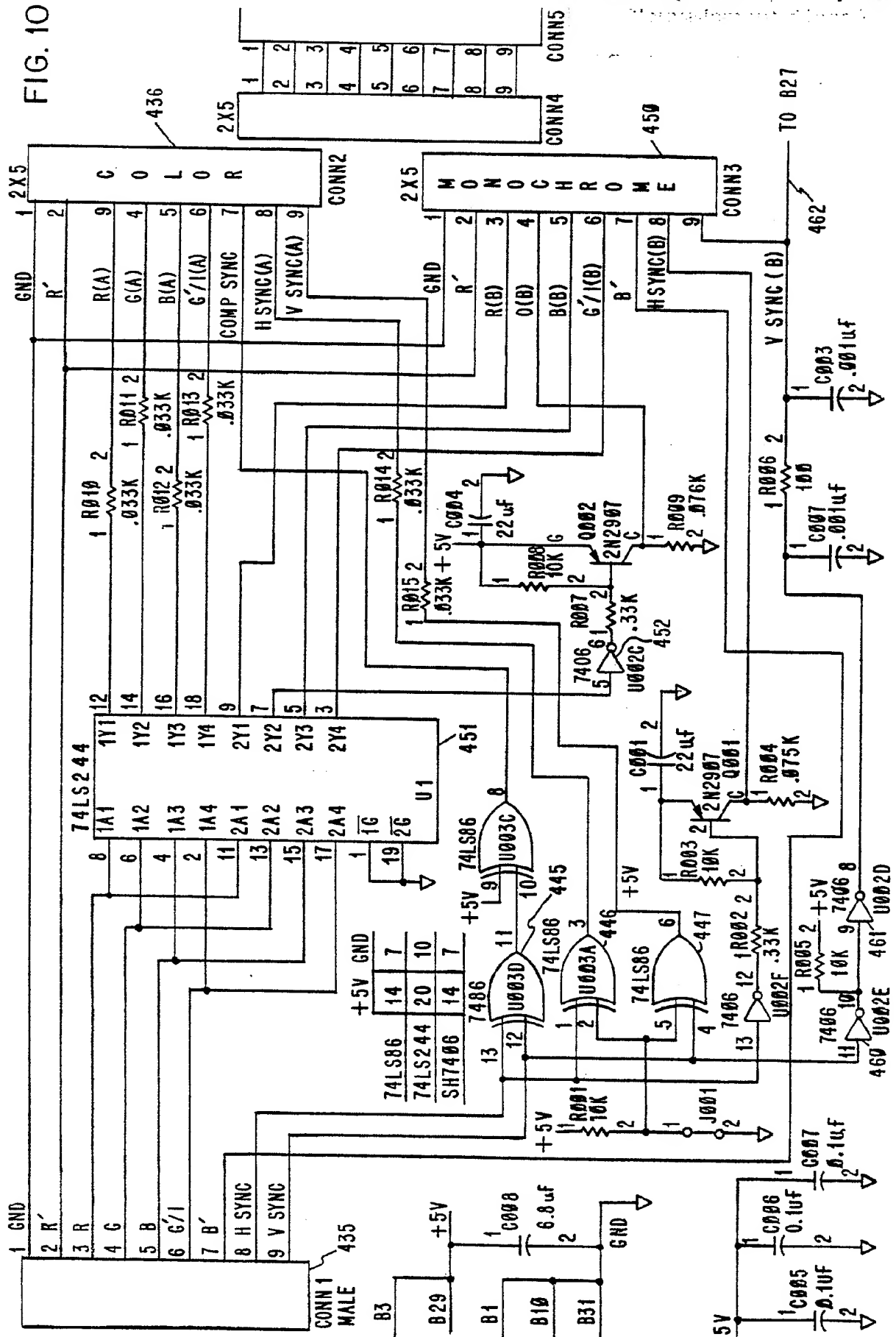


FIG. 9

FIG. 10



GDIS

GDISWORK STATION

FIG. 11

SOFTWARE (GDIS SUPPORT)

- * Vehicle Screen Generator
- Monochrome - Colour
- * Post Processor
- * Host Communication

GDIS

FIG. 12

VEHICLE SCREEN GENERATOR

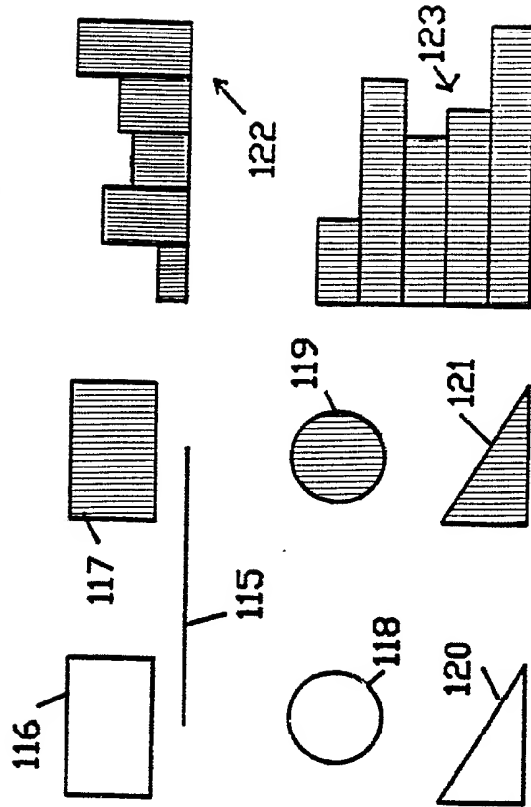
CREATE EDIT FILE

- * Graphic Elements
- * Icons and Symbols
- * Touch Screen Functions
- * Animation

FIG. 13

GDIS

GRAPHIC ELEMENTS



Neu eingereicht / Newly filed
Nouvellement déposé

FIG. 14

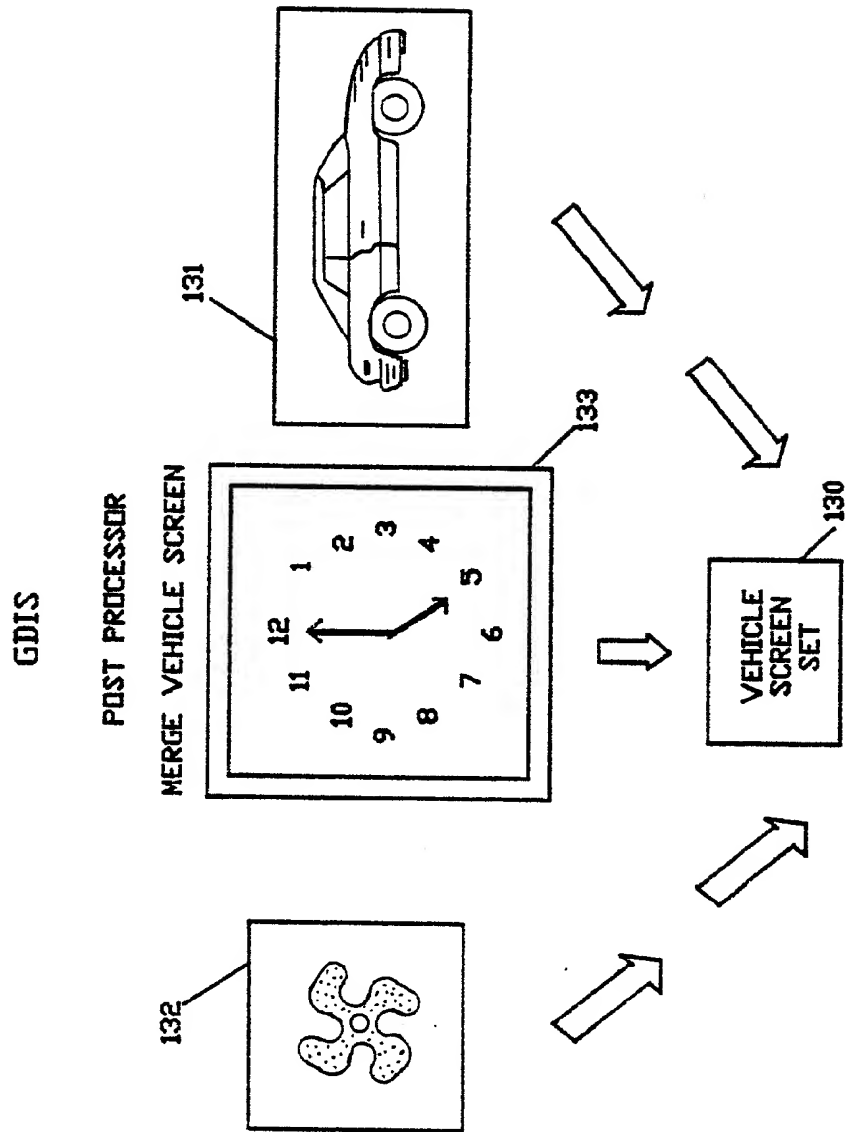


FIG. 15

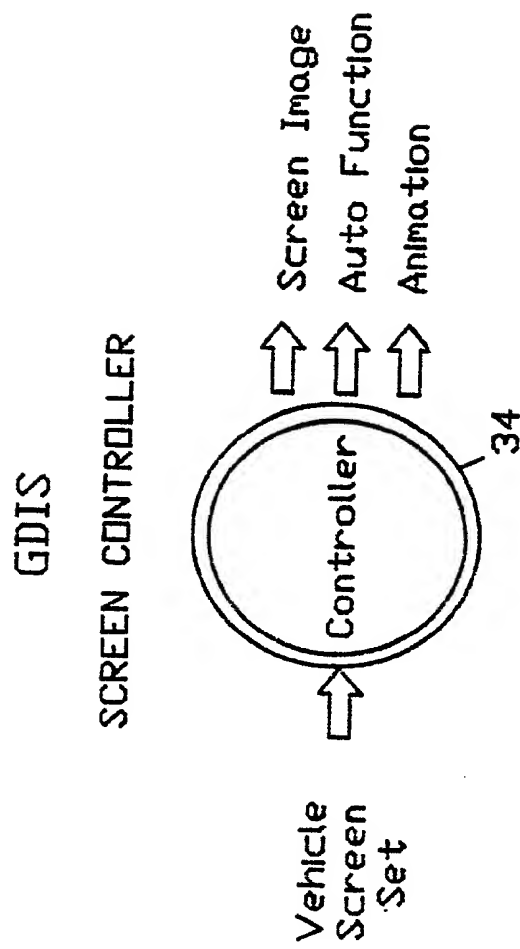


FIG. 16

GDIS
HOST COMMUNICATIONS

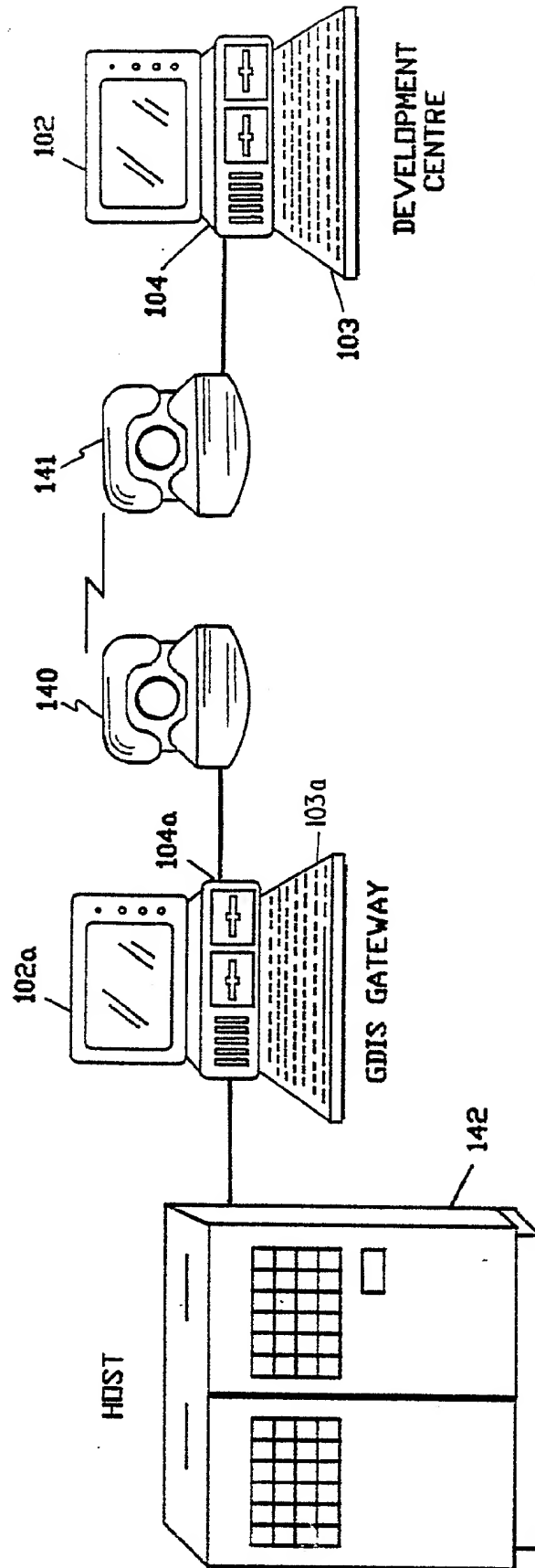
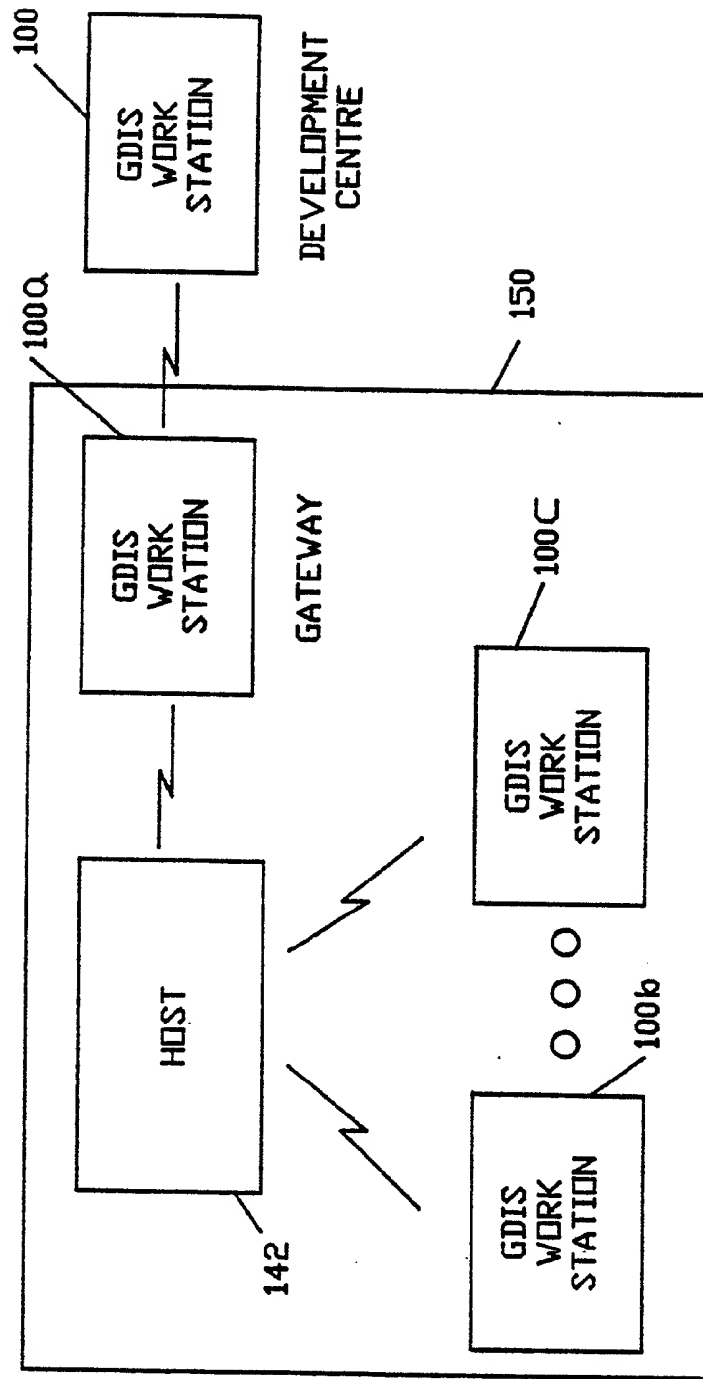


FIG.17

GDIS NETWORK

NETWORK



ALL INFORMATION CONTAINED
HEREIN IS UNCLASSIFIED
DATE 08-01-2001 BY 60322

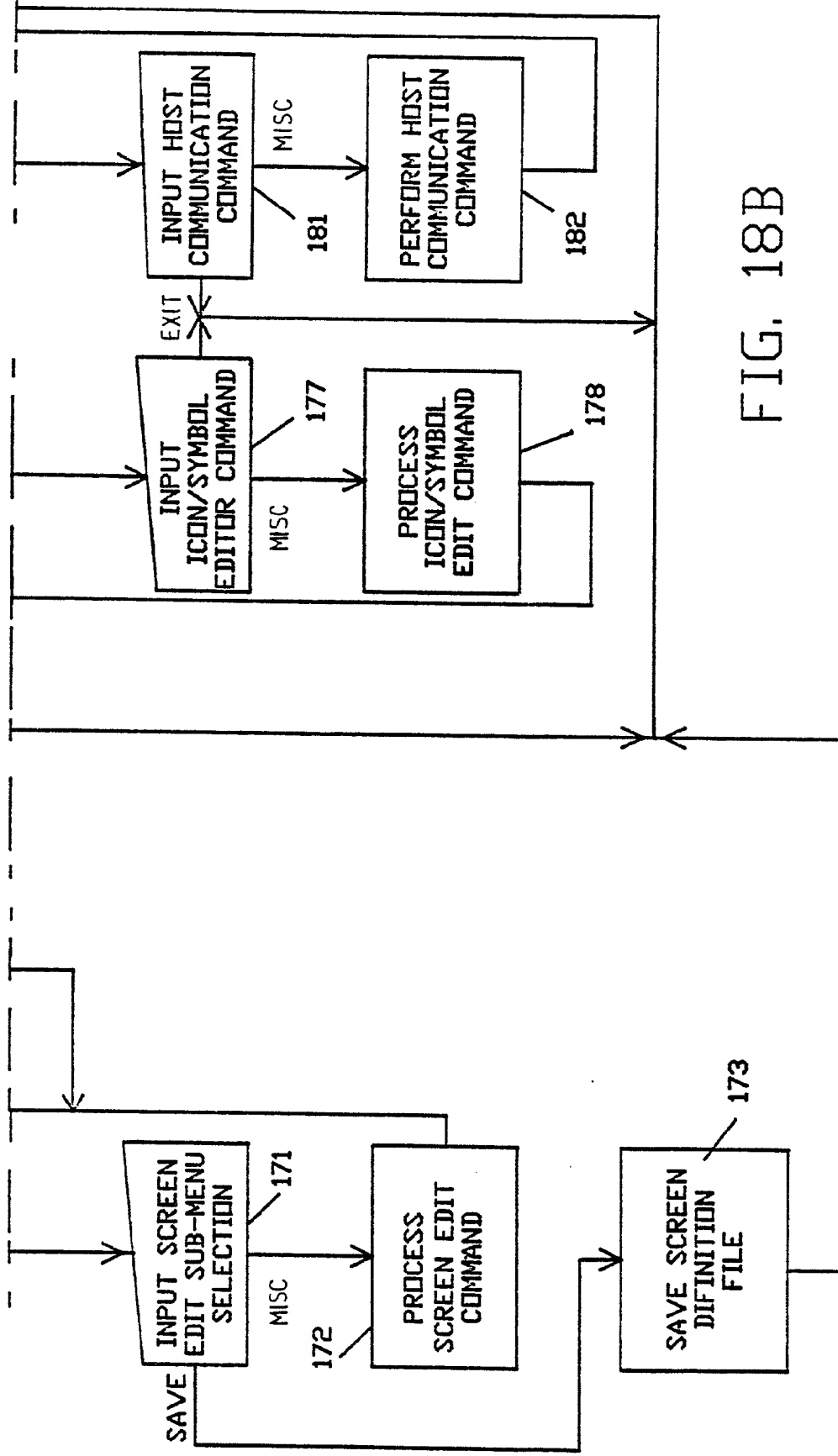


FIG. 18B

GDIS POST-PROCESSOR PROGRAM FLOW

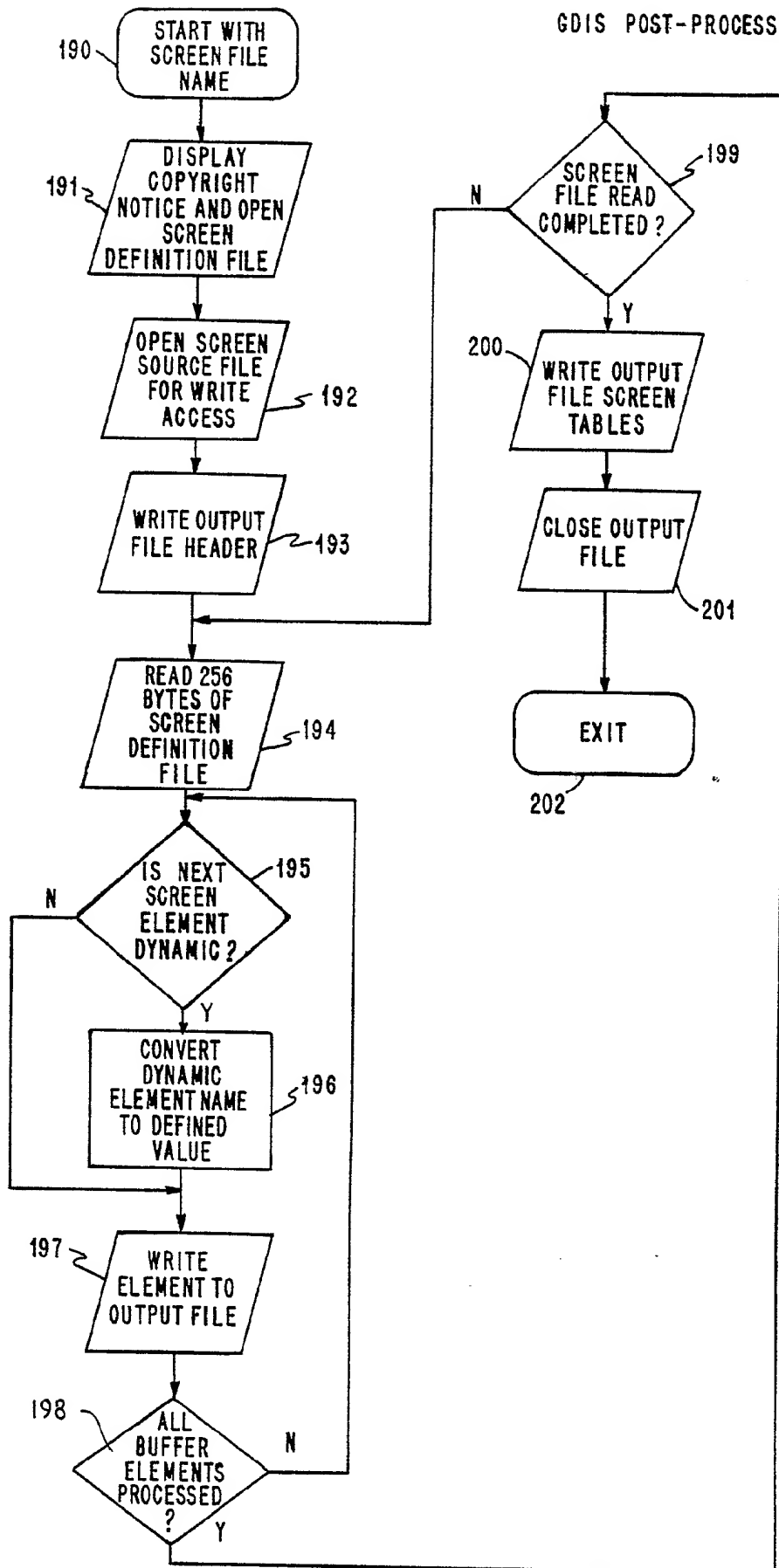
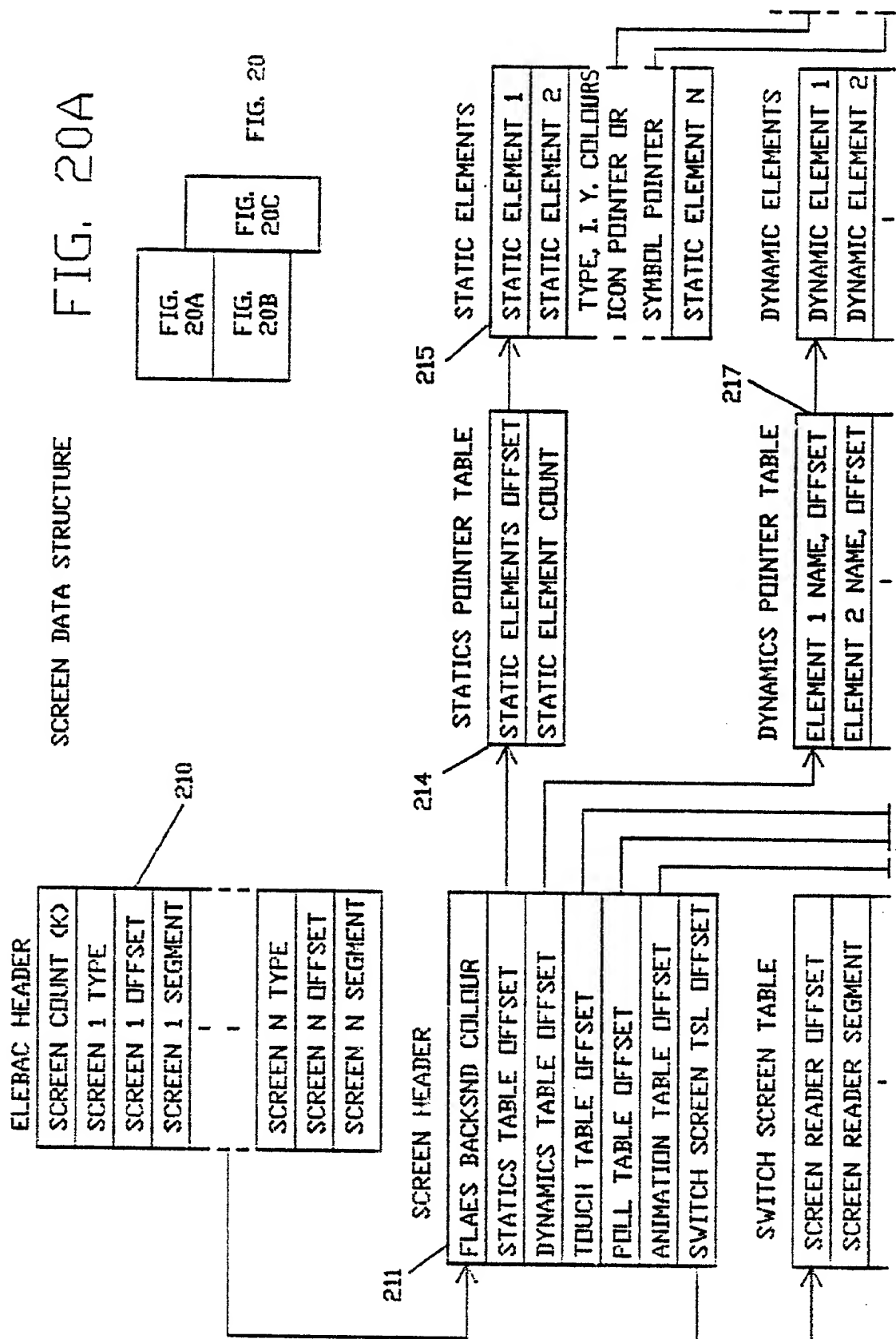


FIG. 19



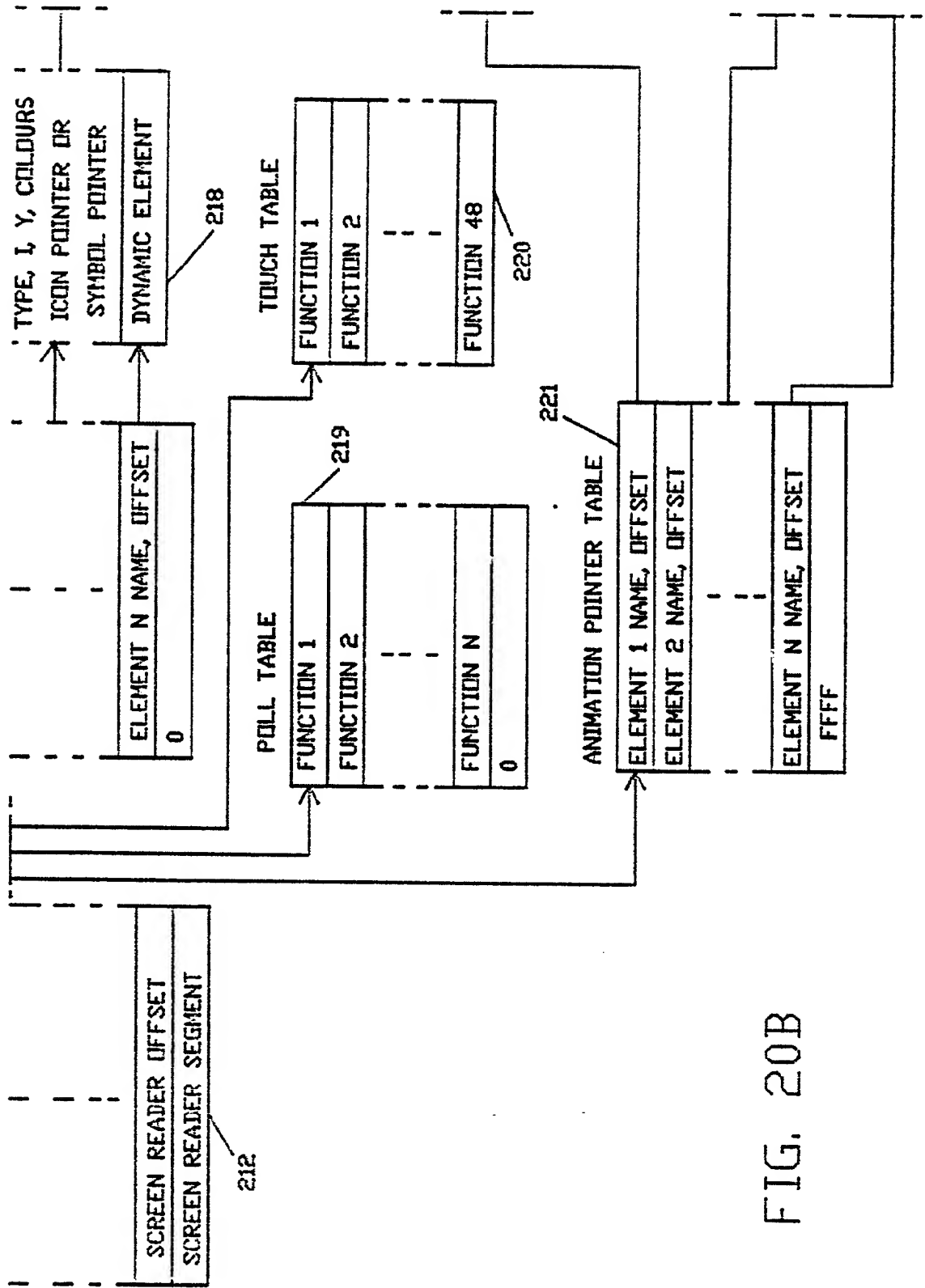


FIG. 20B

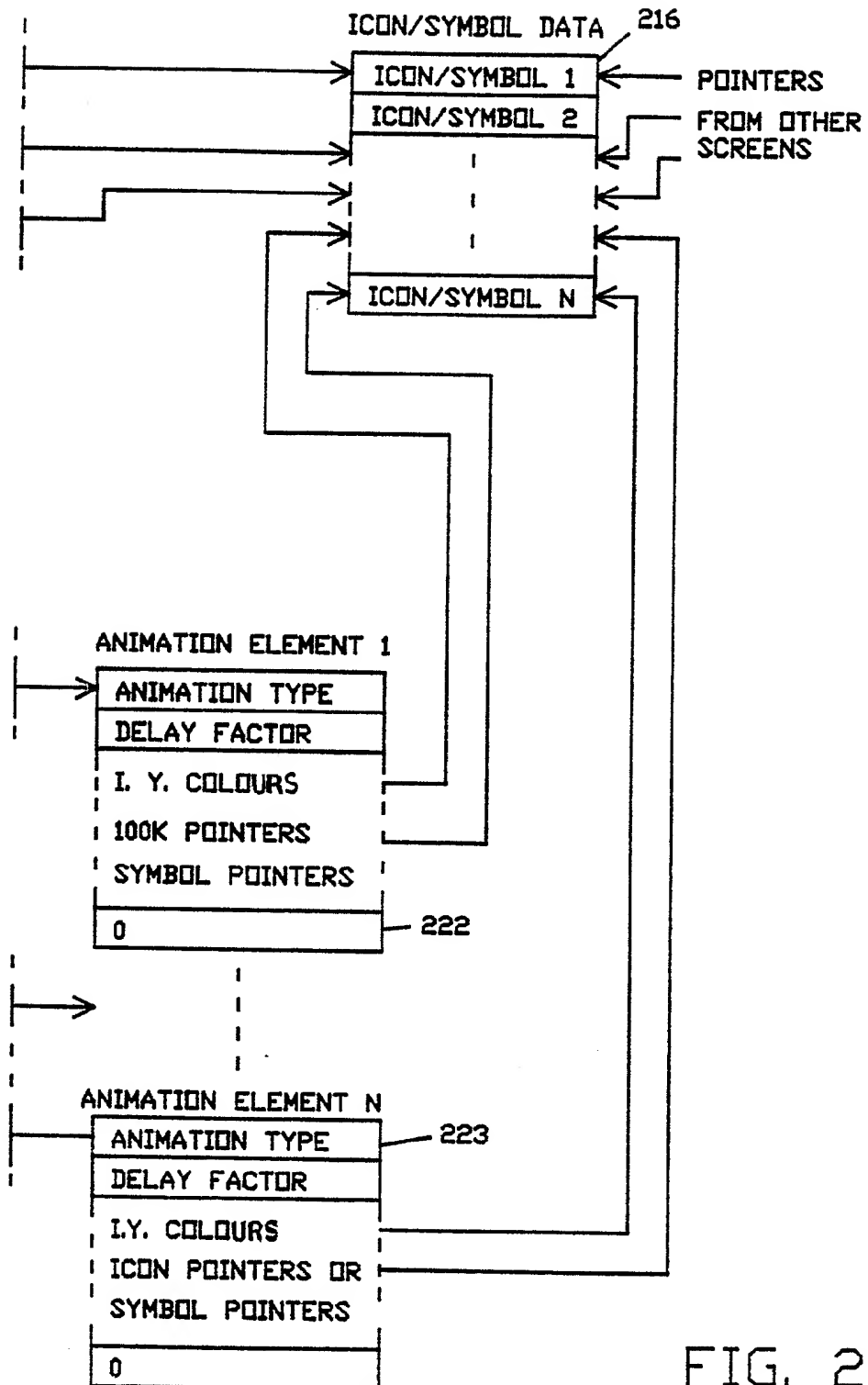


FIG. 20C

TOUCH FUNCTION HANDLER

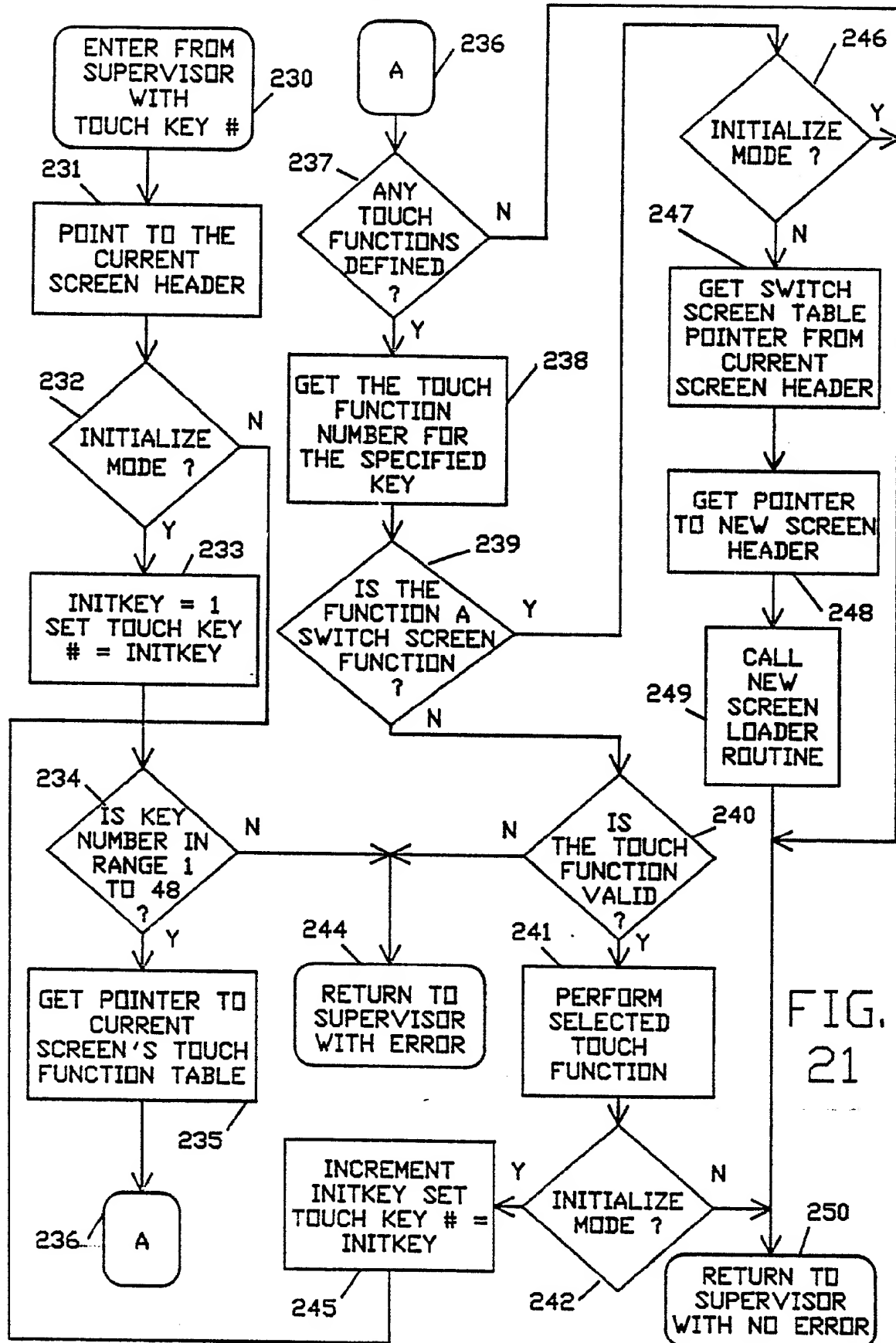


FIG. 21

POLL FUNCTION HANDLER

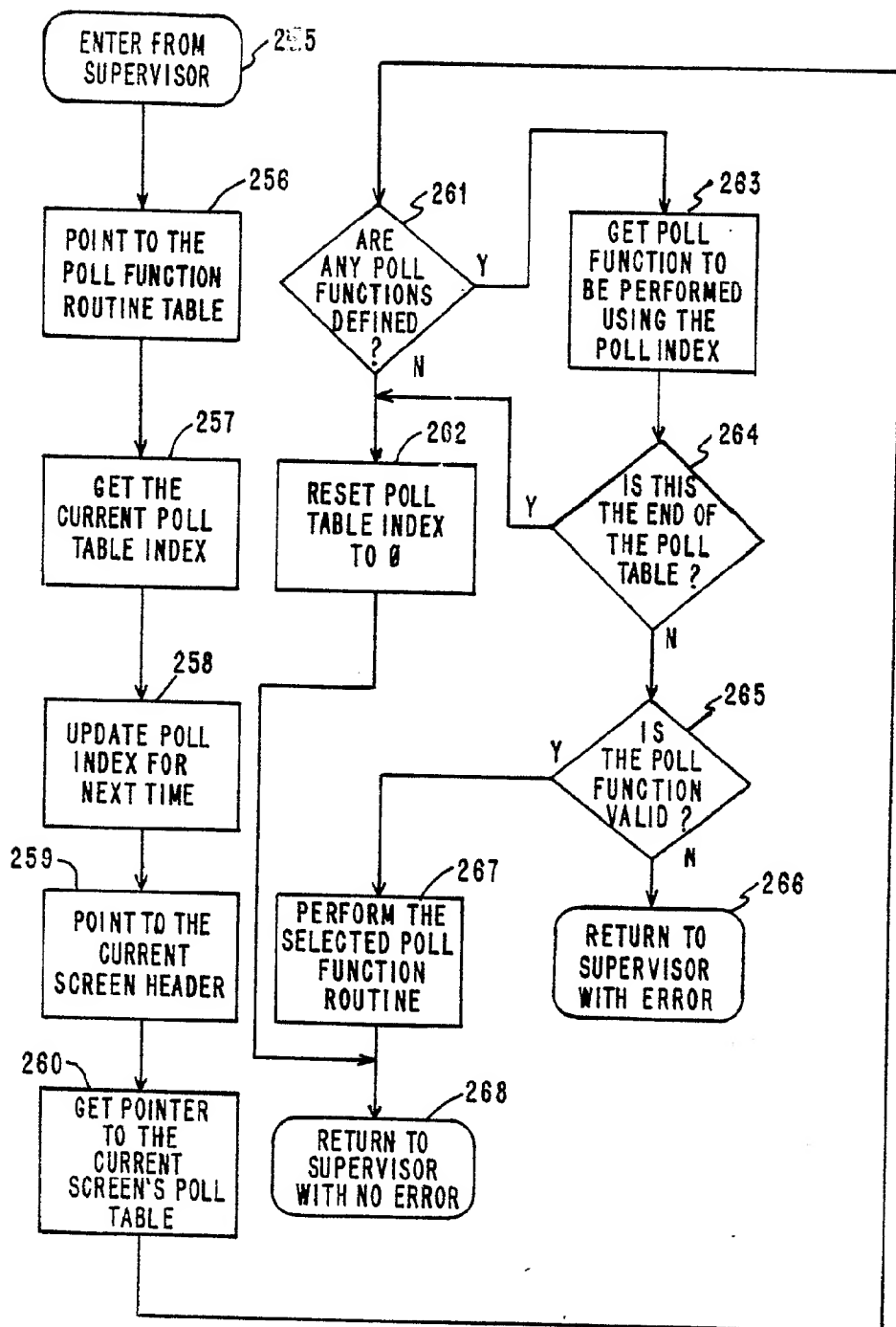


FIG. 22

ANIMATION FUNCTION HANDLER

FIG. 23

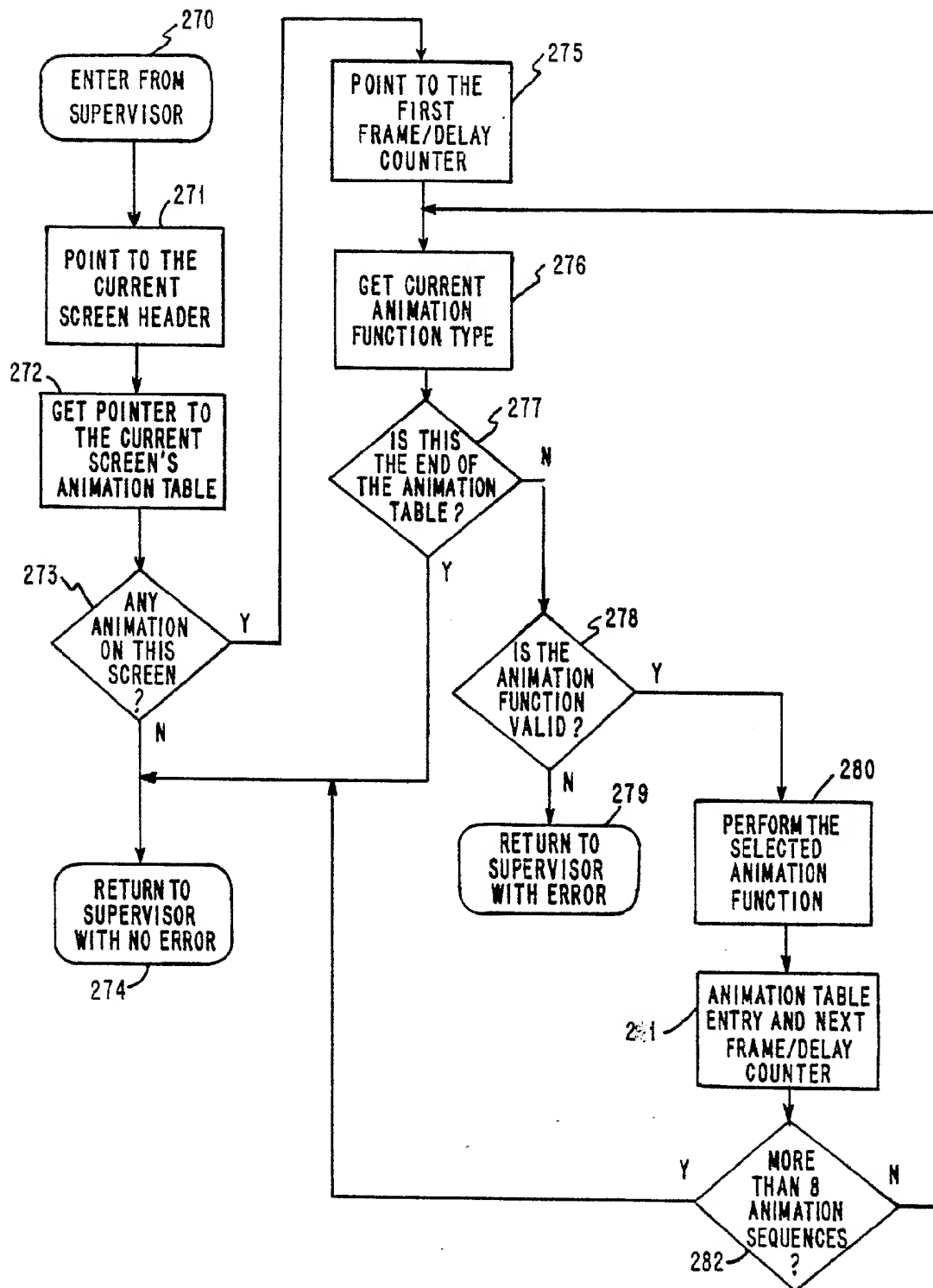


FIG. 24

SCREEN LOADER ROUTINE

